

DESARROLLO DE PROTOTIPO HARDWARE DE UN APUNTADOR BRAILLE CON REPRODUCCIÓN SILÁBICA DE VOZ

Presentado por:

JEAN PIERO DE JESÚS EPALZA RICO
WILLIAM FERNANDO ARENAS VILLABÓN

ING. FREDDYS ENRIQUE VILORIA OBREGÓN

Director

Universidad del Magdalena
Facultad de Ingeniería
Programa de Ingeniería Electrónica
Santa Marta D.T.C.H.
2017

TABLA DE CONTENIDO

TABLA DE CONTENIDO.....	1
1. INTRODUCCIÓN.....	2
2. PLANTEAMIENTO DEL PROBLEMA	4
3. JUSTIFICACIÓN	6
4. OBJETIVOS	7
4.1. Objetivo General	7
4.2. Objetivos Específicos.....	7
5. ALCANCES Y LIMITACIONES	8
6. ANTECEDENTES	10
7. BASE TEÓRICA	13
8. METODOLOGÍA DE ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO	23
9. DESARROLLO DEL PROYECTO	27
10. RESULTADOS.....	48
11. CONCLUSIONES	55
12. BIBLIOGRAFÍA	56
13. ANEXOS	59

1. INTRODUCCIÓN

El Braille es un sistema de lectura y escritura táctil pensado para personas ciegas. Fue ideado por el francés Louis Braille a mediados del siglo XIX. Éste es un sistema universal de celdas de 6 puntos con relieves colocados en dos filas verticales contiguas de tres puntos cada una, ligeramente separados entre sí. El Braille no es un idioma, sino un alfabeto, y con él se pueden representar las letras, los signos de puntuación, los números, la grafía científica, los símbolos matemáticos y la música. (Pérez, 2011).

El presente documento, es el resultado del desarrollo de un prototipo Hardware que contiene elementos electrónicos que permite a las personas ciegas escribir palabras en alfabeto Braille, almacenarlas y posteriormente reproducirlas silábicamente. Así los usuarios pueden estudiar al escuchar los apuntes que se tomaron en Braille.

La metodología que se utilizó para desarrollar este proyecto es la de Estructura de Descomposición del Trabajo o EDT, también conocida por su nombre en inglés Work Breakdown Structure o WBS, es en gestión de proyectos una descomposición jerárquica orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos, con cada nivel descendente de la EDT representando una definición con un detalle incrementado del trabajo del proyecto. La EDT es una herramienta fundamental en la gestión de proyectos (Booz, 2011).

Al ser un prototipo deja las puertas abiertas para que futuros trabajos de grado puedan hacerle mejoras al diseño aquí propuesto y actualizaciones tecnológicas que superen las limitaciones tecnológicas de hoy, de tal forma que permitan mejorar la calidad de vida de la población ciega en Colombia.

CAPÍTULO I: IDENTIFICACIÓN

2. PLANTEAMIENTO DEL PROBLEMA

Colombia es un país en proceso de desarrollo, tanto económico como social; por tal razón la inclusión de las personas con cualquier tipo de limitación es muy poca. Aunque según lo establecido por el congreso de la República de Colombia en la ley estatutaria No. 1618 del 27 de febrero del 2013 (Hernández, 2013), Capítulo IV Artículo once: “DERECHO A LA EDUCACIÓN. El Ministerio de Educación Nacional definirá la política y reglamentará el esquema de atención educativa a la población con necesidades educativas especiales, fomentando el acceso y permanencia educativa con calidad, con un enfoque basado en la inclusión del servicio educativo. Para lo anterior, el Ministerio de Educación Nacional definirá los acuerdos interinstitucionales que se requieran con los distintos sectores sociales, de manera que sea posible garantizar atención integral educativa a la población con discapacidad” Es decir, se quiere garantizar la educación a todas las personas, pero pocas son las herramientas facilitadas para lograrlo cuando se trata de una población con condiciones especiales.

De las diferentes discapacidades que pueden llegar a afectar a una persona, la de mayor interés en este proyecto es la visual.

Según el Censo General 2005 Población con Discapacidad Visual publicado por el DANE en el año 2005, hay un 6.33% de la población total censada con al menos una limitación visual permanente y un 2.73% de personas que necesitan lentes para ver, dando un total de un 9.06% de la población Colombiana con algún tipo de problemas visuales. Pero según una encuesta realizada en el 2014, el 1% de la población (431.042 personas) del país es ciega totalmente; de este número de personas, en el año 2010 solo el 0.2 % de personas invidentes accedieron a la educación en escuelas del estado, según estadísticas del SIMAT.

El Ministerio de Educación Nacional (MEN) en Colombia formuló los estándares básicos de competencias del lenguaje como guía de formación para la educación básica y media, en función de “la importancia que tiene el desarrollo del lenguaje para la formación del individuo y la constitución de la sociedad” (MEN, 2014) Sin embargo, los problemas inherentes a la situación de discapacidad alcanzan a obstaculizar estos objetivos lingüísticos tanto individuales

como sociales planteados por el MEN, situación ante la cual es posible apelar a diferentes entidades públicas del orden nacional, departamental, municipal, distrital y local, en el marco del Sistema Nacional de Discapacidad, son responsables de la inclusión real y efectiva de las personas con discapacidad, debiendo asegurar que todas las políticas, planes y programas, garanticen el ejercicio total y efectivo de sus derechos, de conformidad con el artículo 3° literal c), de Ley 1346 de 2009 (Hernández, 2013).

Para permitir la construcción de conocimiento en un niño con limitación visual, el niño debe desarrollar un nivel de manejo del sistema Braille (inventado por Louis Braille en 1829, mantenido hasta hoy como el código por excelencia para personas con discapacidades visuales); dentro de esta construcción se hace necesario tener en cuenta que la pedagogía de la escritura comienza cuando el conocimiento del alfabeto Braille es satisfactorio, lo cual ocurre alrededor de un mes después de iniciado el aprendizaje de la lectura, pero es clara la necesidad de desarrollar una serie de ejercicios preparatorios para llevar a cabo el proceso de enseñanza – aprendizaje. (Dussán, 2003)

Existen dispositivos y sistemas que intentan facilitar el proceso de estudio de la población invidente, pero su costo es muy elevado como lo son; la Maquina Perkins y Mountbatten Learning System (Balboa, Sánchez, 2012), y no todos pueden acceder a éstos. Algunos incluso guardan la información escrita en Braille para luego ser sintetizada como voz, pero por lo general están desarrollados para el idioma inglés, lo cual representa un problema en un país de habla hispana como lo es Colombia.

En virtud de lo anterior se genera el siguiente interrogante:

- ¿Cómo se puede hacer un Prototipo Hardware que permita recopilar palabras en alfabeto Braille, almacenarlas y posteriormente reproducirlas sonoramente?

3. JUSTIFICACIÓN

En el proceso evolutivo tecnológico de la humanidad, la electrónica se ha convertido en una de las áreas del conocimiento que más aportes ha realizado, brindando soluciones a determinadas problemáticas. Sus múltiples áreas de aplicación, que van desde la industria, y las telecomunicaciones, hasta el área de la pedagogía y educación; desarrollan herramientas y dispositivos con una funcionalidad muy puntual y flexible para las necesidades de cada sector.

A nivel educativo la electrónica ha servido como ente facilitador del acceso a la información, con el desarrollo de la red de Internet (que va muy ligado a las Tecnologías de la Información y Comunicación TICs), capaz de proporcionar cualquier tipo de información, e incluso formación sin importar ni el momento ni el lugar. La electrónica también ha ofrecido una artesanía pedagógica de gran impacto, brindando la posibilidad de inclusión en la formación académica a aquellas personas con discapacidades físicas e incluso mentales y cognitivas. (Prieto).

Las limitantes de comunicación entre personas discapacitadas y no discapacitadas originan la discriminación social y limitan la inclusión social de las primeras. Además, socialmente pocos se preocupan por buscar mecanismos eficientes para interactuar con ellos, produciendo analfabetismo, exclusión laboral y por tanto pobreza (Balboa, Sánchez, 2012).

Se han hecho grandes avances en la tecnología para los usuarios de Braille durante la última década. Lamentablemente, la mayoría de los estudiantes lectores de este sistema en los primeros grados no tienen acceso a gran parte de esta tecnología. Se pretende que este proyecto sea una herramienta de inclusión social y educativa, la cual pueda brindar la posibilidad, a personas con problemas de visión y de escasos recursos económicos, de formarse académicamente (en los niveles básico, medio y de educación superior) de una manera más fácil y amena. También se espera que esta herramienta facilite las actividades realizadas por el personal docente especializado en personas invidentes en las instituciones educativas.

4. OBJETIVOS

4.1 Objetivo General

Desarrollar un Prototipo Hardware que permita la adquisición de palabras en Alfabeto Braille, almacenarlas y luego reproducirlas sonoramente.

4.2 Objetivos Específicos

- Diseñar diagramas y esquemas necesarios para fabricar el prototipo Hardware.
- Almacenar en el prototipo Hardware palabras del lenguaje español utilizando el Alfabeto Braille.
- Reproducir silábicamente las palabras almacenadas.

5. ALCANCES Y LIMITACIONES

En el proceso del proyecto se han dividido los siguientes alcances y limitaciones:

ALCANCES

- Como primera instancia, se pretende desarrollar un Prototipo Hardware de un Apuntador en Alfabeto Braille con reproducción silábica de voz.
- En este proyecto se desea desarrollar un prototipo Hardware que permita a las personas invidentes tomar apuntes de palabras del idioma español de forma electrónica y almacenarlos para poder estudiarlos posteriormente.
- El Prototipo Hardware de un Apuntador Braille con reproducción silábica de voz, tendrá una conexión con puerto Rs232 que permitirá su conexión con computadoras para poder ser utilizado como base de trabajos futuros o para realizarle mejoras al mismo.

LIMITACIONES

- Una de las principales limitantes de este proyecto será la reproducción silábica de todas las palabras, números y caracteres del idioma español, debido a que la memoria de programa del microcontrolador que se usará es limitada y el sistema Braille es demasiado extenso.
- Se optará por trabajar con los componentes electrónicos que se puedan adquirir en el mercado Colombiano para así reducir costos de fabricación, conociendo que se pueden encontrar componentes más eficientes en el mercado exterior, pero con un valor comercial más elevado.

CAPÍTULO II:

MARCO TEÓRICO

6. ANTECEDENTES

Desde el inicio de los tiempos el hombre ha tratado de transmitir a sus futuras generaciones el conocimiento que ha logrado en el transcurso de su vida, con el fin de permitir un desarrollo constante de su sociedad. Hasta la fecha en el departamento del Magdalena, Colombia, no se han presentado proyectos de esta índole, aunque en zonas aledañas existen proyectos similares pero enfocados a diferentes puntos los cuales a continuación se presentan y toman como base para llevar a cabo este proyecto.

En la universidad Manuela Beltrán se desarrolló un trabajo titulado “SISTEMA ELECTROMECHANICO DE LA LECTURA Y LA ESCRITURA EN BRAILLE”, en el cual, después de identificar las necesidades y problemas de accesibilidad al medio que diariamente viven las personas limitadas sensorialmente, el grupo de Aplicabilidad Tecnológica de la Universidad Manuela Beltrán con apoyo de Colciencias - Sena, ha querido colaborar con esta población, diseñando y desarrollando un prototipo que les permita a los niños invidentes entre tres y ocho años, aprender a comunicarse mediante la lectoescritura del sistema Braille de una forma interactiva y lúdica. El prototipo permite al usuario aprender a escribir y leer; letras, sílabas, palabras y frases cortas en el sistema Braille a través de la retroalimentación auditiva de los fonemas escritos.

En la universidad Manuela Beltrán desarrollaron un artículo titulado “DISPOSITIVOS PARA LIMITADOS VISUALES DESARROLLADOS POR EL GRUPO APLICABILIDAD TECNOLÓGICA DE LA UMB” en el que se enfocan en el desarrollo de dispositivos para personas con limitaciones físicas. Ha sido uno de los principales móviles del trabajo investigativo del Grupo Aplicabilidad Tecnológica; como prueba de ello se logra la patente de invención del “SISTEMA PARA EL APRENDIZAJE DE LA LECTURA Y ESCRITURA EN BRAILLE (SLE)”, y la aprobación por parte de COLCIENCIAS - En la convocatoria nacional para la presentación de proyectos de investigación - año 2003 - subdirección de programas de innovación y desarrollo empresarial - de una segunda versión, para “EL SISTEMA ELECTRÓNICO MECÁNICO PARA EL APRENDIZAJE DE LA LECTO-ESCRITURA DEL BRAILLE”, con sus necesarias mejoras tecnológicas. Estos progresos tecnológicos, ayudan al

estudiante limitado visualmente a generar gusto por el aprendizaje de manera fácil y entretenida, características que resultan ser de esencial importancia en cualquier proceso pedagógico, ya que con ellas se asegura efectividad en la formación, y se busca emular el proceso de aprendizaje convencional de lectura y escritura, que realizan los niños, tanto personas con visión normal como invidentes, en forma táctil y audible. Los dos dispositivos se proyectan para representar cada uno de los caracteres del lenguaje tradicional del Braille; empleando una interpretación particular de la matriz Braille o signo generador, para lograr que cada una de las codificaciones se correspondan con cada caracter único de la escritura Braille y que puedan utilizarse como herramientas didácticas que estimulen tanto el sentido auditivo como el sentido del tacto, mediante sus respectivas modificaciones e innovaciones. (Dussán, Jiménez, Hernández, Giraldo, Acosta, 2004)

Dicho proyecto se enfoca en cómo resolver las dificultades de aprendizaje del sistema Braille. Básicamente pretende disminuir los tiempos de aprendizaje y aumentar el número de letras aprendidas por sesión. La solución generada de este trabajo es un prototipo que posee un teclado Braille y genera un pitido cada vez que se escribe correctamente una palabra Braille. La autora manifiesta su interés en que el grupo de trabajo tecnológico de su universidad realice un segundo prototipo que pronuncie las palabras o las letras que conforman la palabra.

En el trabajo de grado “SISTEMA DE ENSEÑANZA DEL CÓDIGO BRAILLE PARA NIÑOS CON LIMITACIONES VISUALES” (Cétares, Cortés y Silva, 2006) de la Universidad Javeriana se describe un dispositivo que ayuda en el proceso de aprendizaje del código Braille, el cual está compuesto por una base de datos de archivos de sonido, un software de acceso, una interfaz electromecánica y una unidad de conversión y procesamiento. (Cétares, Cortés, Silva, 2005)

En la Universidad Distrital Francisco José de Caldas elaboraron un trabajo de grado llamado “DESARROLLO TECNOLÓGICO PARA EL MEJORAMIENTO DE LA COMUNICACIÓN A DISTANCIA ENTRE PERSONAS CON DISCAPACIDAD”, donde su objetivo es proporcionar a la población en general, especialmente a las personas en situación de discapacidad, un dispositivo de comunicación que mediante; un proceso inteligente, un sistema

de codificación/decodificación adaptativo y unas interfaces accesibles, les permita fomentar unas adecuadas competencias comunicativas, como mecanismo de inclusión social, aportándoles las posibilidades de satisfacer adecuadamente algunas necesidades humanas fundamentales. (Jiménez, Hernández, 2009)

En el Instituto Tecnológico De Mérida, elaboraron un trabajo final del taller de investigación denominado “ENTRENADOR BRAILLE ELECTRÓNICO CON INTERFAZ USB DE HARDWARE Y SOFTWARE ABIERTO”. Este proyecto pretende, mediante un aparato electrónico, facilitar la enseñanza del sistema Braille aprovechando tecnologías electrónicas baratas, al alcance y presentes en la vida cotidiana actual, como son las computadoras, con el objetivo de aumentar el número de personas que conozcan y utilicen el sistema Braille. (Balboa, Sánchez, 2012)

En la revista Redes de Ingeniería publicaron un artículo titulado como “PROTOTIPO DE TECNOLOGÍA EN ASISTENCIA PARA LA ENSEÑANZA DEL BRAILLE” con el diseño y fabricación del dispositivo denominado Sistema Electrónico Mecánico para el aprendizaje de la Lectoescritura del Braille (SEMLEB). Junto con la aplicación de una metodología basada en acompañamiento de un tutor, se buscó determinar y analizar los factores que facilitaran el proceso de aprendizaje de la lectoescritura del sistema Braille para niños ciegos en etapa escolar y sus repercusiones en el aprendizaje del idioma español para identificar las letras del alfabeto, los fonemas, las sílabas, y algunas palabras y frases de corta longitud. (Hernández, Jiménez, 2011)

7. BASE TEÓRICA

El sistema Braille fue una invención de Louis Braille, quien a los 15 años de edad creó un sistema de lectura para personas con limitaciones visuales, aprovechando la capacidad de captar por medio de las terminales nerviosas de las yemas de los dedos, el Símbolo Generador o como se le conoce más comúnmente Celda Braille. Este símbolo está formado por seis puntos distribuidos en dos columnas de tres puntos cada una, dependiendo de la presencia o la ausencia de esos puntos, se obtienen 64 combinaciones suficientes para representar varias letras del alfabeto. (Cétares, Cortés, Silva, 2005)

Este sistema está diseñado para ser utilizado a través del tacto, por medio de puntos en relieve. La unidad básica o Signo Generador es el cajetín o celdilla. En este espacio se sitúan los 6 puntos en relieve, distribuidos en dos columnas de tres puntos cada una. Esta celdilla o cajetín, mide aproximadamente unos 5 mm de alto por 2,5 mm de ancho. La distancia horizontal entre celdillas es de unos 6,30 milímetros y la vertical entre líneas es de 10,20 milímetros, aproximadamente. Estas medidas hacen que la información quepa dentro de la yema de un dedo.

Cada letra o signo se representa en un solo cajetín, en el que aparecen o no los 6 puntos en relieve, que son percibidos a través del tacto por las yemas de los dedos. En un texto en Braille, los cajetines no están presentes, siendo visibles sólo los puntos. Para identificar los puntos, se les atribuye un número del 1 al 6. (Instituto de Tecnologías Educativas)

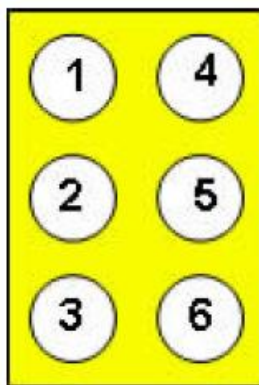


Ilustración 1: Signo Generador Braille, tomado de (Instituto de Tecnologías Educativas)

Mediante las diferentes combinaciones de puntos en un mismo cajetín se pueden obtener 64 formas distintas de disposición de los puntos, incluyendo el cajetín en blanco, que se utiliza para separar las palabras. Como el número de posibilidades es limitado, por economía del sistema, un mismo signo Braille puede significar cosas distintas, según el contexto donde lo utilicemos o si le antepongamos otro signo. Por esta razón, el Braille es un sistema y no un simple alfabeto ya que, utilizando sus 64 combinaciones, se han desarrollado distintos códigos; para matemáticas, ciencias, música, estenografía (Braille abreviado), signografía específica para diferentes idiomas, etc. (Instituto de Tecnologías Educativas)

En Braille, cada letra se representa con una combinación de puntos en relieve. Para dejar espacios en blanco entre palabras, se emplea también un espacio o cajetín en blanco. Al comenzar a escribir se dejan dos espacios en blanco (sangría) y, entre párrafo y párrafo, es conveniente dejar un renglón en blanco, igual que en tinta. Estos espacios son muy útiles para que el lector pueda localizar fácilmente el inicio de cada párrafo y, así, darle facilidades para ubicarse en el texto. (Instituto de Tecnologías Educativas)

El código Braille está diseñado de manera lógica, mediante series:

1ª Serie: Se utilizan únicamente los cuatro puntos superiores (1, 2, 4,5) y con ellos se forman las diez primeras letras del alfabeto:

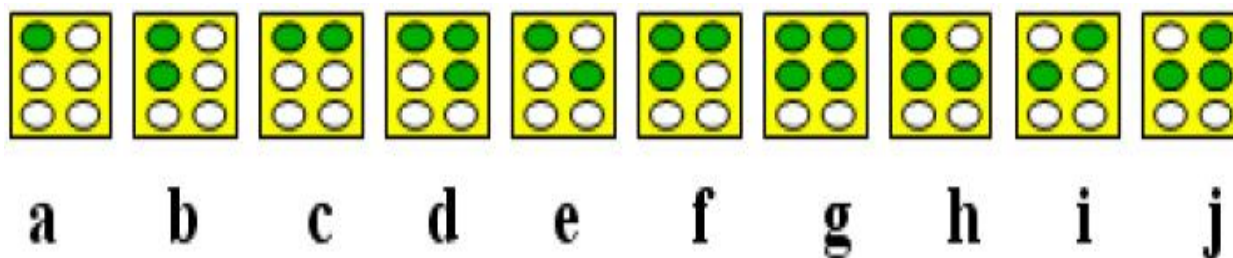


Ilustración 2: Alfabeto Braille desde la letra "a" hasta la letra "j", tomado de (Instituto de Tecnologías Educativas)

2ª Serie: Se forma con los puntos de la primera serie, añadiéndoles el punto número 3 y, así, obtenemos las siguientes letras, a excepción de la letra “ñ” (conviene recordar aquí que Luis Braille era Francés):

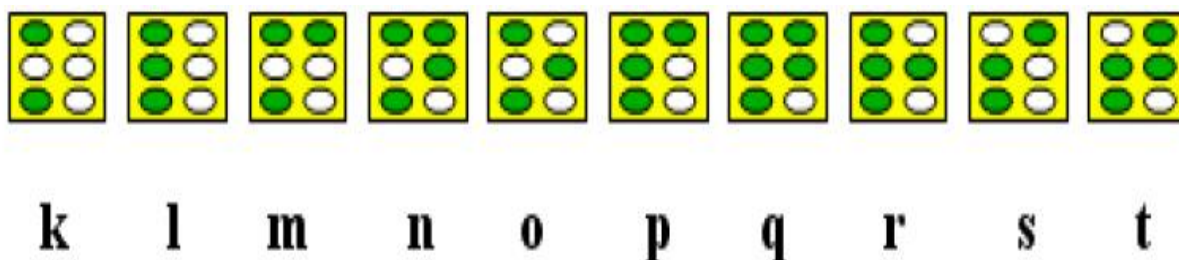


Ilustración 3: Alfabeto Braille desde la letra "k" hasta la letra "t", tomado de (Instituto de Tecnologías Educativas)

3ª Serie: Se forma con los puntos de la serie 2ª, añadiendo el punto número 6:

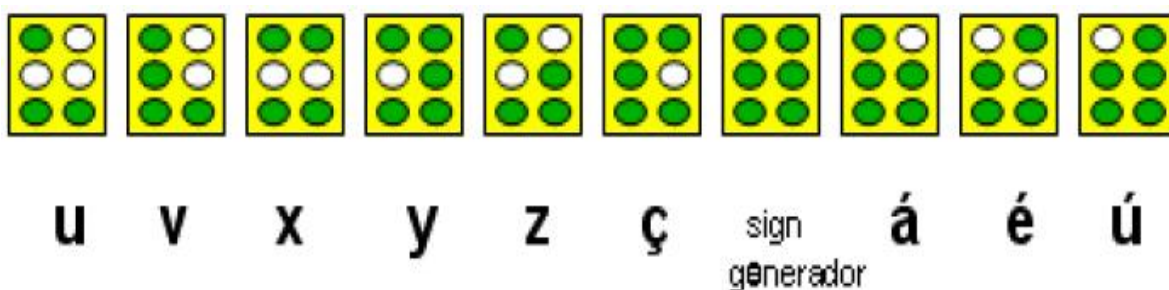


Ilustración 4: Alfabeto Braille desde la letra "u" hasta la letra "z" y vocales "a" "e" "u" acentuadas, tomado de (Instituto de Tecnologías Educativas)

4ª Serie: Son los elementos de la 1ª serie, añadiendo el punto número 6. En esta serie se forman signos propios del francés, por lo que aquí sólo presentamos las letras que nos interesan en español:

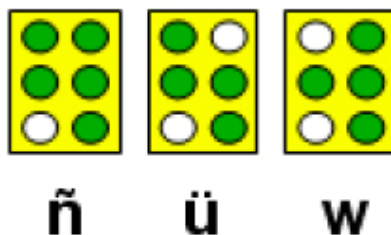


Ilustración 5: Alfabeto Braille letras "ñ" "ü" "w", tomado de (Instituto de Tecnologías Educativas)

5ª Serie: Son los signos de la primera serie, pero utilizando los puntos de la mitad inferior de la celdilla o cajetín. Así, conseguimos los signos de puntuación:

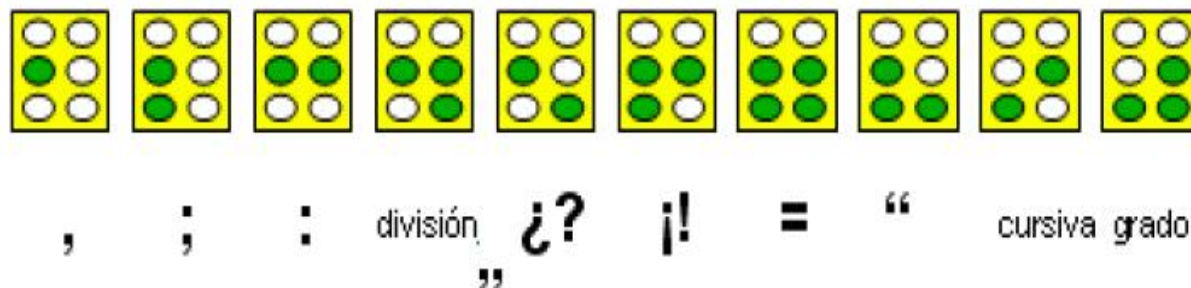


Ilustración 6: Signos de Puntuación en Braille, tomado de (Instituto de Tecnologías Educativas)

Las vocales con tilde se representan mediante estas combinaciones de puntos:

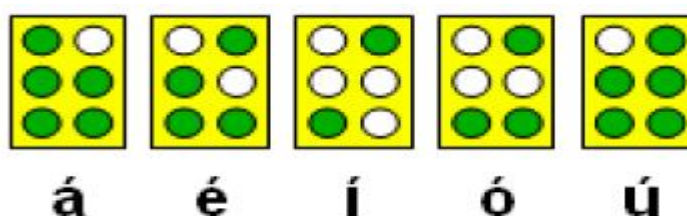


Ilustración 7: Vocales Acentuadas en Braille, tomado de (Instituto de Tecnologías Educativas)

Como las 64 combinaciones posibles son insuficientes para la formación de todos los grafemas necesarios, es necesario utilizar signos complementarios que, antepuestos a una determinada combinación de puntos, convierten una letra en mayúscula, cursiva, número o nota musical:



Ilustración 8: Signos Complementarios en Braille, tomado de (Instituto de Tecnologías Educativas)

Por ejemplo, anteponiendo el signo de mayúscula, formado por los puntos 4 y 6, a cualquier letra, obtenemos las letras mayúsculas:

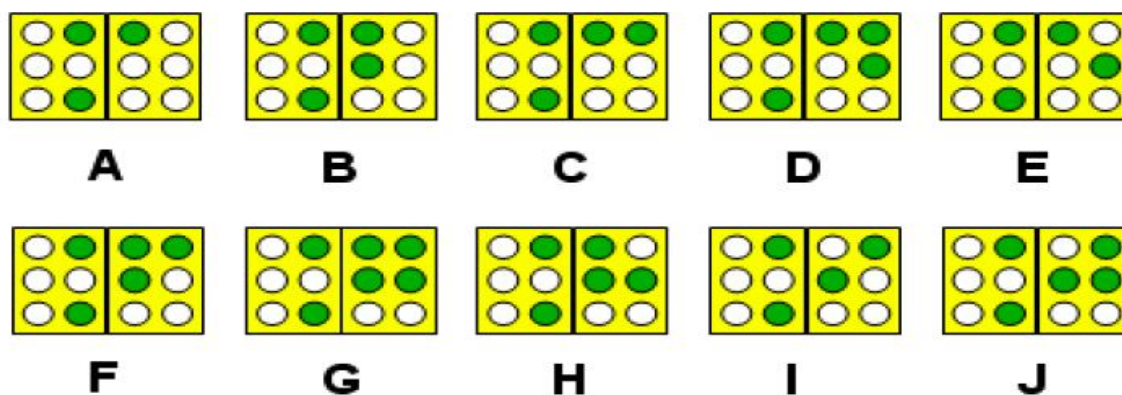


Ilustración 9: Letras Mayúsculas en Braille, tomado de (Instituto de Tecnologías Educativas)

De la misma forma, anteponiendo el signo de número, formado por los puntos 3, 4, 5 y 6, a la primera serie, obtenemos los números del 0 al 9:

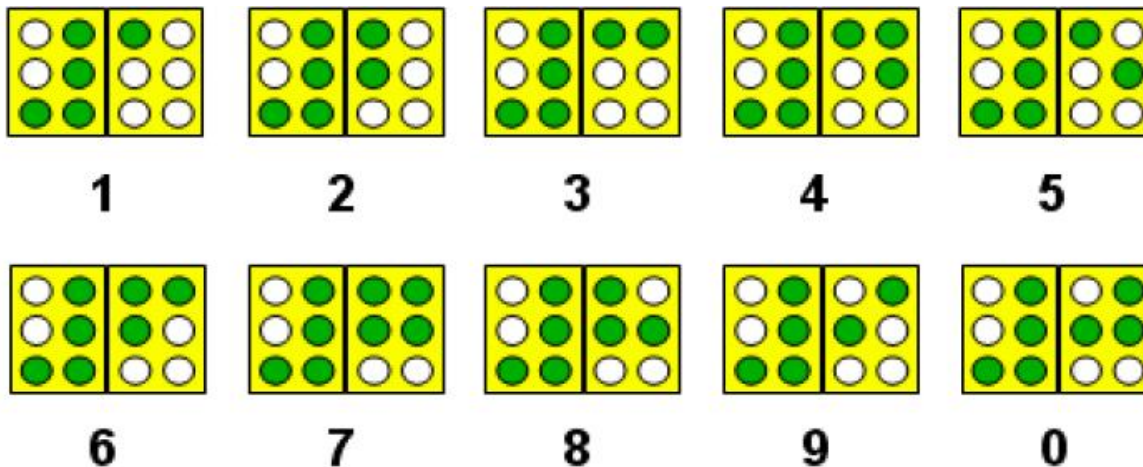


Ilustración 10: Números del 0 al 1 en Braille, tomado de (Instituto de Tecnologías Educativas)

Para cantidades de dos o más cifras o números decimales sólo se coloca el signo de número delante de la primera cifra. La coma decimal se representa con el punto 2. Para números altos puede utilizarse el punto 3, como en vista, para el punto de separación de órdenes de unidades:

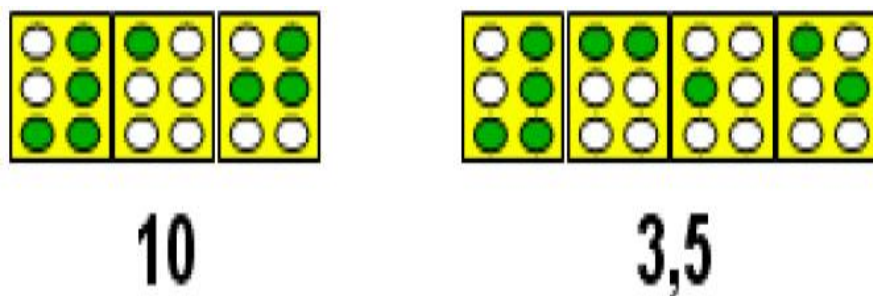


Ilustración 11: Números de dos cifras y números decimales en Braille, tomado de (Instituto de Tecnologías Educativas)

Por último, se muestran a continuación los signos matemáticos básicos:

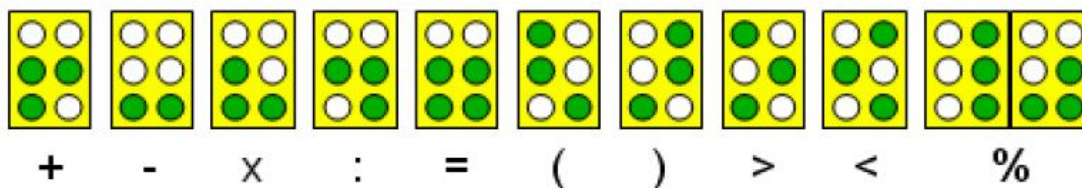


Ilustración 12: Signos Matemáticos Básicos en Braille, tomado de (Instituto de Tecnologías Educativas)

El PIC16F877A es un microcontrolador con memoria de programa tipo FLASH, lo que representa gran facilidad en el desarrollo de prototipos y en su aprendizaje ya que no se requiere borrarlo con luz ultravioleta como las versiones EEPROM, sino que permite reprogramarlo sin ser borrado con anterioridad.



Ilustración 13: PIC 16F877A, tomado de http://www.microchip.com/_images/ics/medium-PIC16F877A-PDIP-40.png

El microprocesador **PIC16F877A** es un producto de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden. (Microchip)

Estos microcontroladores poseen varias características (Ver Ilustración 9) que hacen de él un dispositivo muy versátil, eficiente y práctico para ser empleado en circuitos tales como el que se desarrolla en este proyecto. Sus principales características son:

- Soporta modo de comunicación serial, posee dos pines para ello.
- Amplia memoria para datos y programa.
- Memoria reprogramable: La memoria en este PIC es la que se denomina FLASH; este tipo de memoria se puede borrar electrónicamente.
- Posee un set de instrucciones reducidas (tipo RISC), pero con las instrucciones necesarias para facilitar su manejo.

Nombre del parámetro	Valor
Tipo de memoria de programa	Flash
Memoria de programa (KB)	14
CPU Speed (MIPS)	5
RAM Bytes	368
EEPROM de datos (bytes)	256
Digital Periféricos de Comunicación	1-UART, 1-A/E/USART, 1-SPI, 1-I2C 1-MSSP (SPI/I2C)
Periféricos Captura / Comparación / PWM	2 CCP
Timers	2 x 8 bits, 1 x 16 bits
ADC	8 ch, 10 bits
Comparadores	2
Rango de temperatura (C)	-40-125
Rango de Voltaje de funcionamiento (V)	2-5,5
Número de pines	40

Ilustración 14: Características de funcionamiento de un Microcontrolador PIC16F877A, tomado de <http://www.sigmaelectronica.net/pic16f877a-p-96.html>

ISD4004: Es una alternativa para aplicaciones electrónicas que requieran trabajar con voz. Estos circuitos son útiles tanto para grabar como para reproducir sonidos, funcionan con una alimentación de tres volts, y el tiempo de almacenamiento de sonido está entre 8 y 16 minutos dependiendo de la frecuencia de muestro (ver *Ilustración 15*). (López, Julio, 2005)

Part Number	Duration (minutes)	Input Sample Rate (KHz)	Typical Filter Pass Band (KHz)
ISD4004-08M	8.0	8.0	3.4
ISD4004-10M	10.0	6.4	2.7
ISD4004-12M	12.0	5.3	2.3
ISD4004-16M	16.0	4.0	1.7

Ilustración 15: Características de Funcionamiento del ISD4004 según su Número de Parte, tomado de (López, Julio, 2005)

Estos son dispositivos basados en tecnología CMOS, e internamente poseen; un oscilador, filtros Antialiasing, filtros Smoothing, características de auto silenciador y memorias flash. Además, son de tipo esclavo, los cuales fueron diseñados para ser utilizados en sistemas basados en microcontroladores o microprocesadores. Tanto el direccionamiento como el control se realizan a través de una interfaz serial periférica (SPI). Las grabaciones son almacenadas en la memoria flash, con la característica de cero poder de consumo de almacenamiento. Las señales de voz y audio son almacenadas directamente en su forma natural. (López, Julio, 2005)

Memoria EEPROM 24LC256: Las memorias EEPROM (Electrically-Erasable Programmable Read-Only Memory) que funcionan bajo el protocolo I2C han ganado poco a poco un espacio en el hardware de los equipos electrónicos hasta llegar al punto de transformarse en uno de los medios de almacenamiento de información más populares por su practicidad y sencillez de manejo, ya que tener la posibilidad de almacenar datos de diversa índole en una memoria no volátil, es una característica importante de los equipos porque permite la desconexión prolongada de cualquier suministro energético y conservar, durante mucho tiempo, información valiosa que de otro modo se perdería al desconectar un sistema. (Cisneros, 2010)



Ilustración 16: Memoria EEPROM 24LC256, tomado de http://www.microchip.com/_images/ics/medium-24LC256-PDIP-8.png

CAPÍTULO III:

DISEÑO METODOLÓGICO

8. METODOLOGÍA DE ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO

La metodología que se utilizó para desarrollar este proyecto es la de Estructura de Descomposición del Trabajo o EDT, también conocida por su nombre en inglés Work Breakdown Structure o WBS, y es en gestión de proyectos, una descomposición jerárquica orientada al entregable del trabajo a ser ejecutado por el equipo de proyecto para cumplir con los objetivos de éste y crear los entregables requeridos, con cada nivel descendente de la EDT representando una definición con un detalle incrementado del trabajo del proyecto. La EDT es una herramienta fundamental en la gestión de proyectos (Booz, 2011).

El propósito de una EDT es organizar y definir el alcance total aprobado del proyecto según lo declarado en la documentación vigente. Su forma jerárquica permite una fácil identificación de los elementos finales, llamados "Paquetes de Trabajo". Se trata de un elemento exhaustivo en cuanto al alcance del proyecto. La EDT sirve como base para la planificación del proyecto. Todo trabajo a realizarse en el proyecto debe poder rastrear su origen en una o más entradas de la EDT (Nasa, 2001).

Es por todo lo anterior que se utiliza dicha metodología para el desarrollo de este proyecto, debido a que permite la libertad de dividir de manera flexible los trabajos y las tareas a realizar, ya que permite:

- Distribuir los Objetivos del proyecto en actividades.
- Fraccionar las actividades en tareas.

Fase I: Diseñar el prototipo Hardware.

- **Recopilación de Datos:** Se realizó una documentación referente a los proyectos similares realizados a nivel mundial y en otras regiones a nivel nacional.
- **Análisis de la Información:** Se clasificó la información más relevante obtenida en la actividad anterior.

➤ **Diseño de las Etapas:** Aquí se plantean las diferentes etapas que va a tener el prototipo, el cual consta de cuatro de ellas que son; la etapa de entrada, etapa de almacenamiento, etapa de procesamiento y la etapa de salida.



Ilustración 17: Etapas en las que se descompone el Proyecto

❖ **Etapas de Entrada:** Este módulo permite la escritura y lectura de los datos en código Braille

❖ **Etapas de Almacenamiento:** En esta etapa se guardan los datos del código Braille.

❖ **Etapas de Procesamiento:** Aquí se procesan todos los datos que fueron almacenados en el módulo de Almacenamiento.

❖ **Etapas de Salida:** Tiene la finalidad de reproducir silábicamente lo almacenado y procesado en los módulos anteriores.

Fase II: Almacenamiento

➤ **Análisis de Herramientas:** Para desarrollar este proyecto se estudió la variedad de recursos tecnológicos de acuerdo a su funcionalidad, características y costos; y se decidió qué dispositivos se implementarían en cada etapa, de la siguiente manera:

❖ **Etapas de Entrada:**

- **ISD4004-16M:** A través del micrófono recibe y almacena la voz. Posteriormente la reproduce. Dicho integrado se maneja con el microcontrolador PIC16F877A.

- **Celda Braille:** Conjunto de 6 botones cuya función es tomar las letras en código Braille.

- **Botones de Configuración:** Un botón para grabar y otro botón para reproducir.

❖ **Etapas de Almacenamiento:**

- Memorias EEPROM 24LC256

Fase III: Reproducir silábicamente las palabras almacenadas.

➤ **Análisis de Controladores y Dispositivos:** Para desarrollar las etapas de procesamiento y salida se analizaron la diversidad de controladores y dispositivos existentes en el mercado de acuerdo a su funcionalidad, características y costos; y se escogieron la siguiente forma los componentes para cada una de estas dos etapas:

❖ **Etapas de Procesamiento:**

- Microcontrolador PIC16F877A

❖ **Etapas de Salida:**

- **Parlante:** Transductor para transformar señal eléctrica a señal sonora.
- **Amplificador de Audio:** Configuración en puente para la salida del parlante de 4 Ohms/1W.
- **Zumbador:** Señal sonora que indica algunos procesos que se efectúan en el prototipo.
- **LEDs:** Señales visibles, algunos indican funciones que se ejecutan y otros sirven para la configuración del prototipo.

Fase IV: Base para futuros proyectos similares

- **Evaluación:** En esta etapa se realizan diferentes pruebas al prototipo, para determinar la confianza y facilidad que ofrece la utilización de un Apuntador Braille con Reproducción Silábica de Voz, permitiendo establecer ajustes en los diseños elaborados en las fases anteriores.
- **Recomendación:** En esta etapa se realizan las recomendaciones para trabajos futuros por si algún estudiante de la Universidad del Magdalena desea seguir y mejorar este proyecto.

9. DESARROLLO DEL PROYECTO

➤ Descripción del prototipo por etapas

El sistema sobre el cual se desarrolló el Prototipo Hardware de un Apuntador Braille con Reproducción Silábica de Voz, básicamente consta de 4 etapas que se describen a continuación:

❖ **Etapas de Entrada:** Etapa en la que ingresan al prototipo todos los datos necesarios para poder cumplir con los objetivos del proyecto, tales como; grabar y reproducir las sílabas requeridas para formar palabras del idioma español, recibir datos o apuntes que desea grabar el usuario, conformar las sílabas que se reproducirán a la salida y por último, interrupciones del prototipo que permiten al usuario realizar funciones específicas. Esta etapa está conformada por los siguientes dispositivos:

- **ISD4004-16M:** Este dispositivo es el que permite realizar las grabaciones (a través de un micrófono previamente instalado) de las 942 (Ver Anexo 1) sílabas necesarias para formar palabras del idioma español, las cuales son la base para cumplir los objetivos de este proyecto. Este integrado electrónico, también es el que posibilita las reproducciones en sílabas de la voz real. Su funcionamiento es controlado por un microcontrolador PIC16F877A previamente instalado y configurado.



Ilustración 18: ISD4004-16M instalado en el Prototipo

○ **Celda Braille:** Se encuentra conformada por 6 pulsadores con alto grado de sensibilidad, enumerados cada uno (respetando la configuración real del alfabeto Braille). Dichos botones cuentan con una distribución en forma de matriz 3x2 (3 filas y 2 columnas), con distancias de separación de 1,8 centímetros entre cada uno de ellos.



Ilustración 19: Celda Braille instalada en el exterior del Prototipo

○ **Botones de Grabación y Reproducción:** Se instalaron 2 interruptores normalmente abiertos (pulsadores) quienes dan la orden de realizar las funciones de grabación y reproducción. El pulsador que permite realizar las grabaciones se encuentra etiquetado (exteriormente) como “G” mientras que el botón que habilita las reproducciones silábicas se denomina “R”. Estos interruptores se identifican en el circuito interno como “TeclaRB7” y “TeclaRB6” respectivamente.



Ilustración 20: Interruptores que permiten la Grabación y la Reproducción en el prototipo

○ **Interruptores de Configuración:** El PIC16F877A y las dos memorias EEPROM 24LC256 están siempre energizadas, sin embargo, los demás circuitos no siempre lo estarán, ya que éstos consumen mucha más energía y si el sistema se alimenta a baterías, su tasa de descarga sería mucho mayor. Por tal motivo, se recurre a la instalación de 2 interruptores, para energizar a voluntad el circuito que permite la reproducción de voz conformado por el ISD y el amplificador de audio. Los diferentes cambios de estado de estos interruptores son los que permiten realizar cada una de las funciones del prototipo. Básicamente, el primer switch, identificado como “S-PC” es el que energiza el circuito del MAX232, quien permite establecer comunicación con el computador, mientras que el switch “S-R” permite la energización del circuito del ISD para realizar el proceso de reproducción.

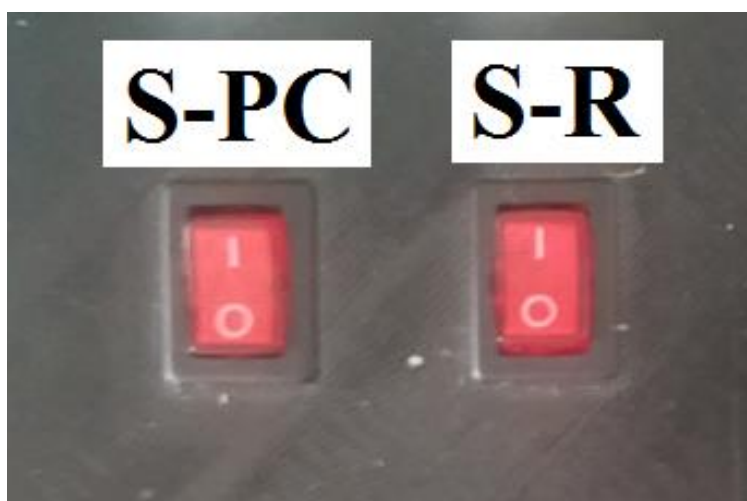


Ilustración 21: Interruptores de Configuración instalados en el Prototipo

○ **Conectores de Tornillo:** En la tarjeta del prototipo, se encuentran instalados 8 conectores de tornillo (2 pines cada uno), en los cuales 6 de ellos van conectados adecuadamente a los 2 pines de cada botón de la matriz Braille, mientras que los 2 conectores restantes se comunican con los Interruptores “G” y “R”.



Ilustración 22: Conectores de Tornillo instalados en el Prototipo

○ **Micrófono:** El micrófono es el dispositivo por el cual se realizan las grabaciones en voz real de los 942 sílabas que se graban en el ISD4004-16M. Su implementación en el prototipo es de forma portable, ya que se puede conectar y desconectar al prototipo por medio de una entrada auricular. Dicha conexión se efectúa alámbricamente.



Ilustración 23: Conexión para el Micrófono instalada en el Prototipo

❖ **Etapas de Almacenamiento:** En esta etapa se realiza el proceso de almacenamiento de los datos que ingresan al prototipo por medio de 2 memorias **EEPROM 24LC256**, una de ellas es extraíble de forma similar a una memoria USB, y la otra se encuentra fija en el circuito impreso.

○ **Memoria EEPROM Extraíble:** En esta memoria se almacenan los apuntes que ingresa el usuario por medio de la Celda Braille. Su instalación es de tipo extraíble similar a una memoria USB. Gracias a este tipo de conexión, el usuario en medio de un proceso de escritura Braille puede reemplazar su memoria inicial por otra nueva sin tener que borrar los apuntes hechos en el proceso, debido a que cada memoria EEPROM tiene su capacidad de almacenamiento limitada, y si se trata de un texto extenso, no podría continuar con la escritura de todos los apuntes deseados.



Ilustración 24: Memoria EEPROM24LC256 Extraíble instalada en el Prototipo

○ **Memoria EEPROM Fija:** Esta memoria contiene las reglas que permiten la descomposición de las palabras en sílabas, y almacena las palabras que se vayan descomponiendo. Otra de sus funciones es servir como memoria de almacenamiento alterno del PIC16F877A para así poder guardar y consultar más información, ya que la memoria de programa de él no es suficiente para almacenar tanta información que se genera al descomponer las palabras en sílabas.

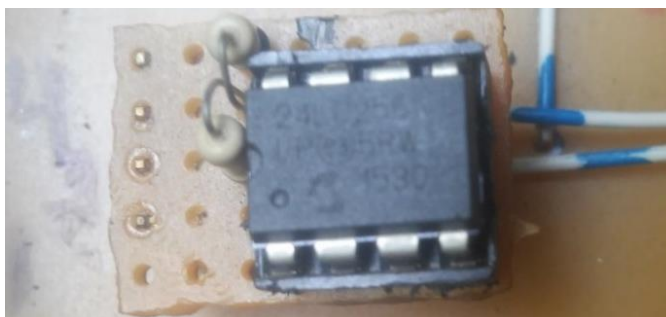


Ilustración 25: Memoria EEPROM 24LC256 Fija instalada en el Prototipo

❖ **Etapas de Procesamiento:** En esta fase del proyecto, se realiza el control de toda la lógica de su funcionamiento, y su principal dispositivo es un microprocesador PIC16F877A.

- **PIC16F877A:** El Prototipo Hardware de un Apuntador Braille con Reproducción Silábica de Voz, tiene como cerebro un microcontrolador PIC16F877A el cual contiene la lógica del programa, el Firmware, (ver Anexo 2). Este microprocesador consta de 40 pines, de los cuales se utilizaron 34 de ellos para su programación, y el resto se dejaron disponibles para las mejoras futuras que se le realicen a este prototipo.



Ilustración 26: Microcontrolador PIC16F877A instalado en el Prototipo

La configuración del PIC16F877A se realizó por medio de un código en ASSEMBLER (ANEXO 2). Dicho Firmware se compone de las siguientes acciones:

- ✓ Registro del conteo de mensajes grabados y reproducidos
- ✓ Comparación y clasificación de letras y números
- ✓ Comunicación con el computador
- ✓ Control del ISD
- ✓ Descomposición de las palabras en sílabas
- ✓ Conversión de texto a voz
- ✓ Control de las 2 memorias EEPROM 24LC256
- ✓ Manejo del teclado de la Celda Braille

❖ **Etapas de Salida:** En esta etapa del proyecto, se encuentran dispositivos que; permiten realizar el proceso de reproducción, establecer comunicación con la computadora y visualizar procesos de ejecución del prototipo. Estos dispositivos se describen a continuación:

- **MAX232N:** Dispositivo electrónico que permite la comunicación con la computadora. Su instalación solo se realiza con el fin de dejar el prototipo como base para trabajos futuros o para que estudiantes de la Universidad del Magdalena le realicen mejoras.



Ilustración 27: MAX232N instalado en el Prototipo

- **Parlante SP-264:** Transductor de 4Ohms/1W que convierte la señal eléctrica en señal sonora. Por medio de éste, se realiza el proceso de la reproducción silábica, el cual tiene una conexión en paralelo con una salida auricular.



Ilustración 28: Parte trasera del Parlante instalado en el Prototipo



Ilustración 29: Salida Auricular instalada en el Prototipo

○ **Amplificador de Audio (Amplificador en Puente):** En esta fase del proceso, se realiza un circuito que permite controlar el volumen y la tonalidad de la voz previamente grabada en el ISD y que se reproduce en la última etapa del proyecto a través del parlante de 40hms/1W. Este circuito se basa en una “configuración en puente” de 2 integrados LM388N-1 (Amplificadores de Audio), tal cual se muestra en la *Ilustración 30*.

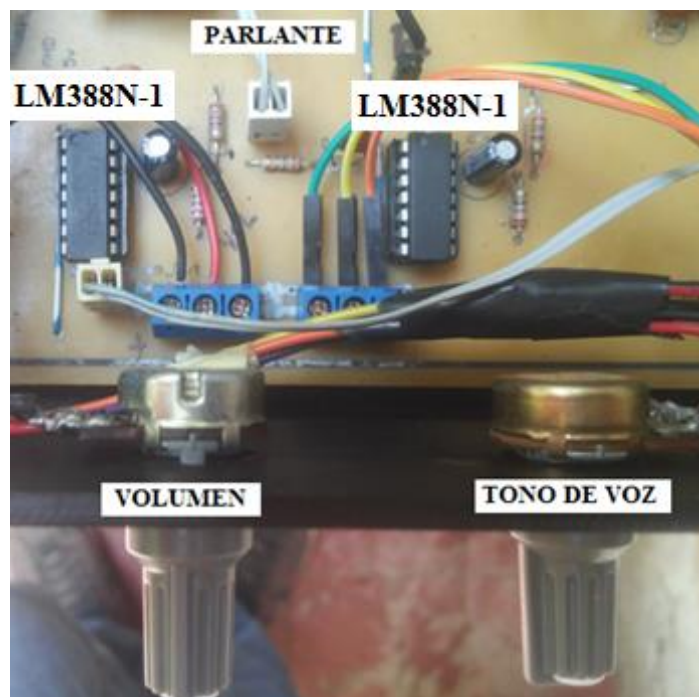


Ilustración 30: Amplificador de Audio - Configuración en puente usada en el prototipo

○ **Zumbador:** Si al energizar el prototipo, los 2 Interruptores de Configuración se encuentran en circuito cerrado al mismo tiempo, el PIC se queda en un ciclo cerrado emitiendo un sonido muy agudo, el cual se genera a través de un Zumbador Piezoeléctrico común. Para eliminar el pitido, se debe apagar el prototipo y abrir uno o los dos switches. Este dispositivo, también se activa para indicar cuando la memoria que almacena los datos (EEPROM extraíble) se llena y no permite seguir con el proceso de almacenamiento. Su descripción se detalla a continuación:

1	Frecuencia resonante	4800±500Hz
2	Voltaje de funcionamiento	Vp-p 1~25
3	Corriente clasificada	Max.5mA, en la onda cuadrada 9Vp-p del deber de 4.8KHz el 50%
4	Salida de sonido en el 10cm	85dB mínimo, en la onda cuadrada 9Vp-p del deber de 4.8KHz el 50%
5	Capacitancia del zumbador piezoeléctrico	el 10000±30% PF en 1KHz
6	Temperatura de funcionamiento	-20°C~+70°C
7	Almacene la temperatura	-30°C~+80°C
8	Peso neto del zumbador piezoeléctrico	0.5g aproximado
9	RoHS del zumbador piezoeléctrico	Si

Ilustración 31: Especificaciones Piezoeléctricas del Zumbador, tomado de <http://spanish.cn buzzer.com/sale-1306798-abs-micro-wire-buzzer-9-volt-piezo-transducer-d14mm-h4-0mm.html>



Ilustración 32: Zumbador Piezoeléctrico instalado en el Prototipo

- **LEDs de Configuración:** Se implementaron para facilitar el proceso de configuración del prototipo, ya que indican las diversas funciones que se van ejecutando.

- **LED ON/OFF (L0):** Como su nombre lo indica, es el que indica cuando el prototipo se encuentra encendido o apagado.

○ **LED 1 (L1):** Este diodo emisor de luz indica varias funciones del prototipo:

- En qué momento empezar a grabar con voz cada sílaba en el ISD.
- Cuando se activa la comunicación con el computador.
- Cuando se ha efectuado el proceso de formateo de la memoria EEPROM extraíble.
- Cuando se ingresa un dato por medio de la Celda Braille.
- Cuando se reproduce una sílaba.

○ **LED 2 (L2):** Indica el momento en el que se activa el proceso de reproducción de las sílabas. Además, avisa cuando en la computadora se empiezan a leer los datos almacenados en la memoria extraíble.

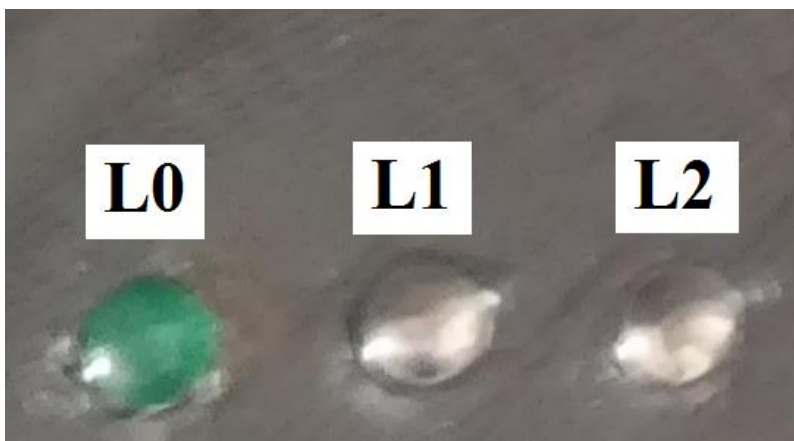


Ilustración 33: LEDs Indicadores instalados en el Prototipo

➤ **Diagrama de Flujo del Funcionamiento:** El Prototipo Hardware de un Apuntador Braille con Reproducción Silábica de Voz, presenta un Diagrama de Flujo de la siguiente forma, la cual se encuentra dividida por etapas:

❖ **Etapas 1:** Se da inicio al programa. Además, se realiza el proceso de formateo de la memoria EEPROM Extraíble y se define qué función es la que se efectuará en el prototipo:

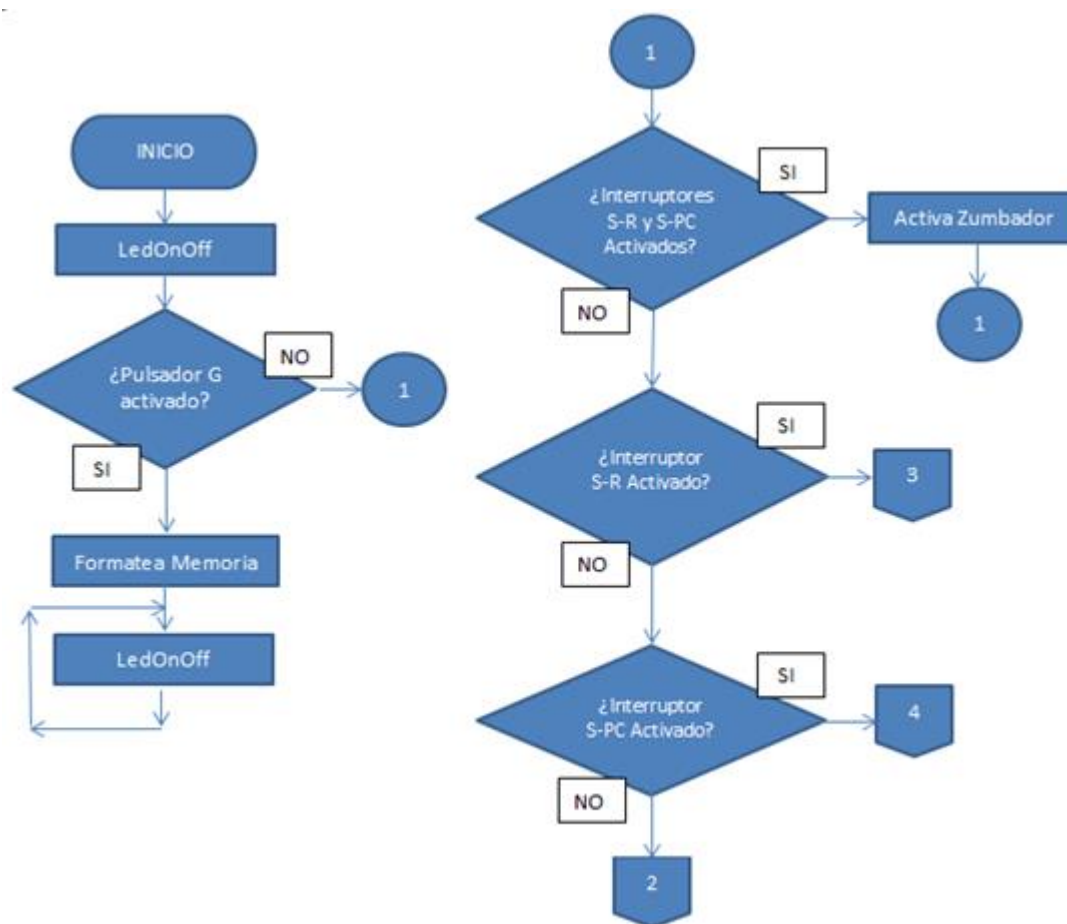


Ilustración 34: Diagrama de Flujo de la Etapa 1 del Funcionamiento del Prototipo

❖ **Etapa 2:** Se presenta todo el proceso de escritura en Braille, y posteriormente se almacena dicha información en la memoria EEPROM Extraíble. Al ingresar a esta etapa del flujo, se encuentra con un ciclo cerrado, del cual se puede salir solo reiniciando el prototipo:

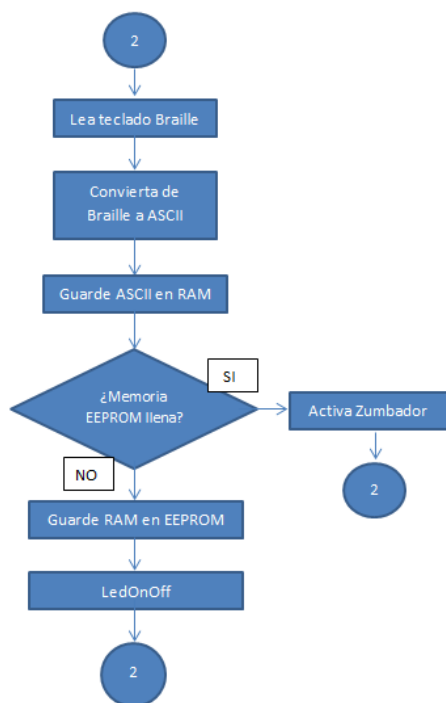


Ilustración 35: Diagrama de Flujo de la Etapa 2 del Funcionamiento del Prototipo

❖ **Etapa 3:** Básicamente es la etapa encargada de todo el proceso de reproducción silábica de los datos ingresados en Braille:

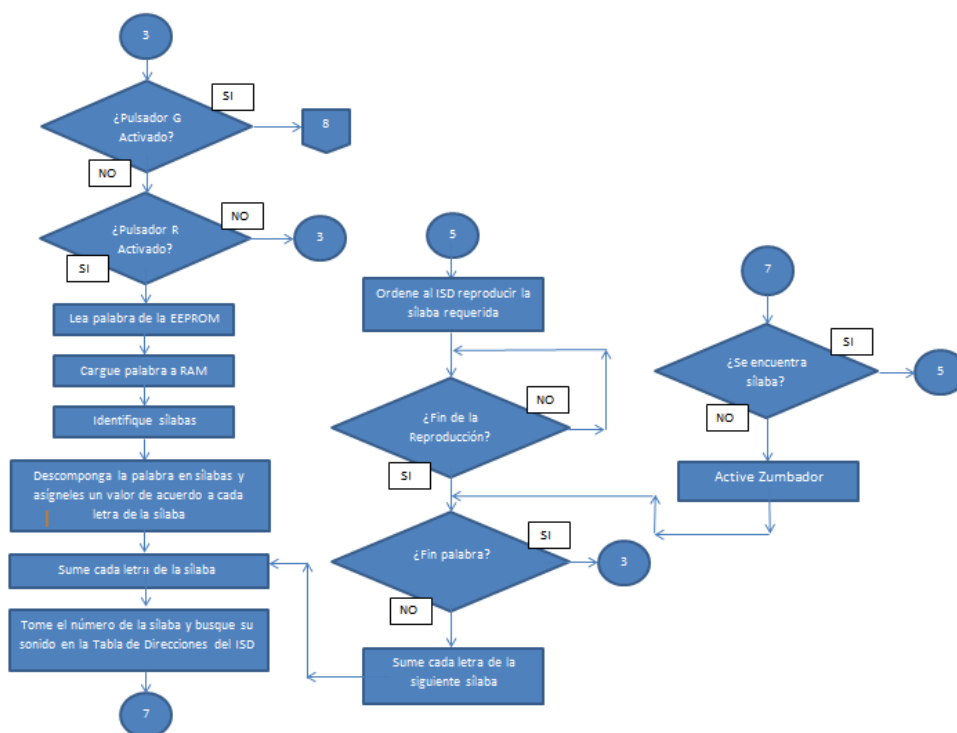


Ilustración 36: Diagrama de Flujo de la Etapa 3 del Funcionamiento del Prototipo

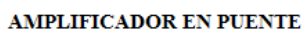


Ilustración 40: Diagrama Esquemático de Amplificador en Puente del Prototipo

➤ **Manual de Funcionamiento del Prototipo Hardware de un Apuntador Braille con Reproducción Silábica de Voz.**

❖ **Formatear la memoria EEPROM Extraíble:** Si la memoria extraíble está llena de información e impide almacenar más datos en ella, o si la memoria es nueva, se debe realizar el proceso de formateo de la misma. Para ello se deben seguir los siguientes pasos:

- Corroborar que el prototipo esté totalmente apagado.
- Mantener presionado el pulsador “G”.
- Encender el prototipo.
- Esperar 7 segundos y luego soltar el pulsador “G”.
- Esperar alrededor de 15 segundos y apagar el prototipo.

La tarjeta del prototipo demora alrededor de 15 segundos para formatear la memoria extraíble. El proceso de formateo borra toda la información que se encuentre almacenada en la EEPROM Extraíble.

❖ **Tomar apuntes en Braille:** Se deben realizar los siguientes pasos en el mismo orden en que se enlistan:

- Asegurarse que el prototipo se encuentre apagado.
- Abrir los dos Interruptores de Configuración S-PC y S-R.
- Encender el prototipo.
- Utilizar el teclado Braille usando los códigos que reconoce el prototipo para cada letra y número (*Tablas 1,2 y 3 - Resultados*).
- Al terminar de escribir en Braille, apagar el prototipo.

Cada vez que se apaga el prototipo durante el proceso de escritura en Braille, el mismo aparato reconoce que terminó el proceso de anotación y va enumerando secuencialmente las notas (Nota 1, Nota 2, Nota 3, ..., Nota 98). El límite máximo de

notas a tomar son 98; si hay más de 90 notas, cada vez que se empiece una nueva nota (es decir desde la Nota 90 hasta la Nota 98), se activa el zumbador una vez indicando que la memoria está casi llena. Si hay 98 notas, la tarjeta se queda pitando en un ciclo cerrado indicando que la memoria está llena y no se puede guardar más información en la memoria extraíble. En este caso, se debe formatear la memoria extraíble o cambiarla por una vacía.

Puede darse el caso que se graben apuntes muy extensos, y que la memoria extraíble se llene antes de llegar a la Nota 98, si ésto sucede, la tarjeta se queda en un ciclo cerrado de la alarma sonora indicando que la memoria está llena y no se puede guardar más información del teclado Braille. La cantidad total de caracteres que se pueden grabar en la memoria extraíble es de 30 mil.

❖ **Reproducir Silábicamente los Apuntes:** Para que el prototipo empiece a leer los apuntes de la memoria extraíble, se debe hacer lo siguiente:

- Asegurarse que la tarjeta esté apagada
- Cerrar el interruptor S-R y abrir S-PC
- Encender el prototipo
- Esperar 5 segundos
- Oprimir el pulsador “R”

Al oprimir el botón “R”, automáticamente la el prototipo empieza a leer desde la Nota1 y, una vez que termina de leerla, continúa con la Nota 2 y así sucesivamente hasta reproducir la última Nota grabada.

Si se desea avanzar entre Notas, se debe oprimir (durante 3 segundos) el botón “1” de la celda Braille. Este proceso interrumpe la reproducción actual y reproduce la siguiente Nota. Se puede repetir tantas veces como se quiera hasta llegar a la Nota deseada.

Una vez terminada la reproducción, se debe apagar el prototipo

❖ **Grabar Sílabas en el ISD:** Una forma de personalizar el prototipo es haciendo que las notas grabadas se reproduzcan con la voz original del usuario. Para ello se habilita la función de Grabación de Voz, y se realiza llevando a cabo el siguiente procedimiento:

- Conectar el micrófono al Prototipo.
- Verificar que el prototipo esté apagado.
- Cerrar el interruptor S-R y abrir S-PC.
- Encender el prototipo.
- Esperar 5 segundos.
- Grabar la sílaba manteniendo presionado el botón “G” hasta pronunciarla completamente.
- Repetir el paso anterior para grabar cada una de las 942 sílabas que se requieren para formar palabras del idioma español. Estas sílabas se deben grabar en el orden indicado y sin saltarse alguna (*Anexo I*).
- Apagar el prototipo cuando se termine de grabar todas las sílabas.

Cuando se mantiene presionado el botón “G” se prenderá el led L1 durante medio segundo, luego se apagará durante otro medio segundo, y posteriormente se volverá a encender y es ahí cuando se debe ingresar la sílaba en voz alta y lo más cercano al micrófono sin soltar la presión sobre el botón “G”. Una vez se termine de pronunciar esa sílaba, se debe soltar el botón “G”, entonces se apagará el led L1 y la sílaba se guardará automáticamente en el ISD.

Se debe tener presente que cuando se deja de oprimir el botón “G” durante el proceso de grabación, el ISD4004-16 guarda inmediatamente el sonido recibido hasta ese punto, es decir, si se suelta el pulsador “G” antes de terminar de pronunciar la sílaba, ésta quedará grabada incompletamente.

Si se llega a equivocar con alguna de las sílabas, se debe apagar el prototipo y empezar de nuevo con el proceso de grabación.

Una vez se termine exitosamente el proceso de grabación de las sílabas, el prototipo reproducirá con la voz personal del usuario todos los apuntes que en él se graben.

❖ **Descargar datos de la EEPROM Extraíble a una computadora:** Para descargar apuntes al computador, se debe realizar el siguiente procedimiento:

- Verificar que el prototipo esté apagado.
- Cerrar el Interruptor S-PC y abrir S-R.
- Conectar el conversor USB-RS232 al puerto RS-232 del prototipo y a la computadora.
- Abrir y configurar el Hyperterminal del computador (9600, sin bit de paridad).
- Encender el prototipo.
- Escribir en el Hyperterminal la orden “*L” (Sin comillas). Luego el prototipo le enviará a la computadora la palabra “OK” y ahí se empezará el proceso de descargue, a la computadora, de todo el contenido de la memoria EEPROM extraíble.
- Cuando el prototipo termine de enviar todos los Bytes de la EEPROM Extraíble, se envía la palabra “FIN”, indicando que el proceso de descargue terminó.
- Apagar el prototipo

CAPÍTULO IV: RESULTADOS

10. RESULTADOS

Se logró desarrollar un Prototipo Hardware de un Apuntador que permite; la escritura en formato Braille, almacenar los datos que allí se ingresan, y reproducir por medio de voz real toda la información guardada.

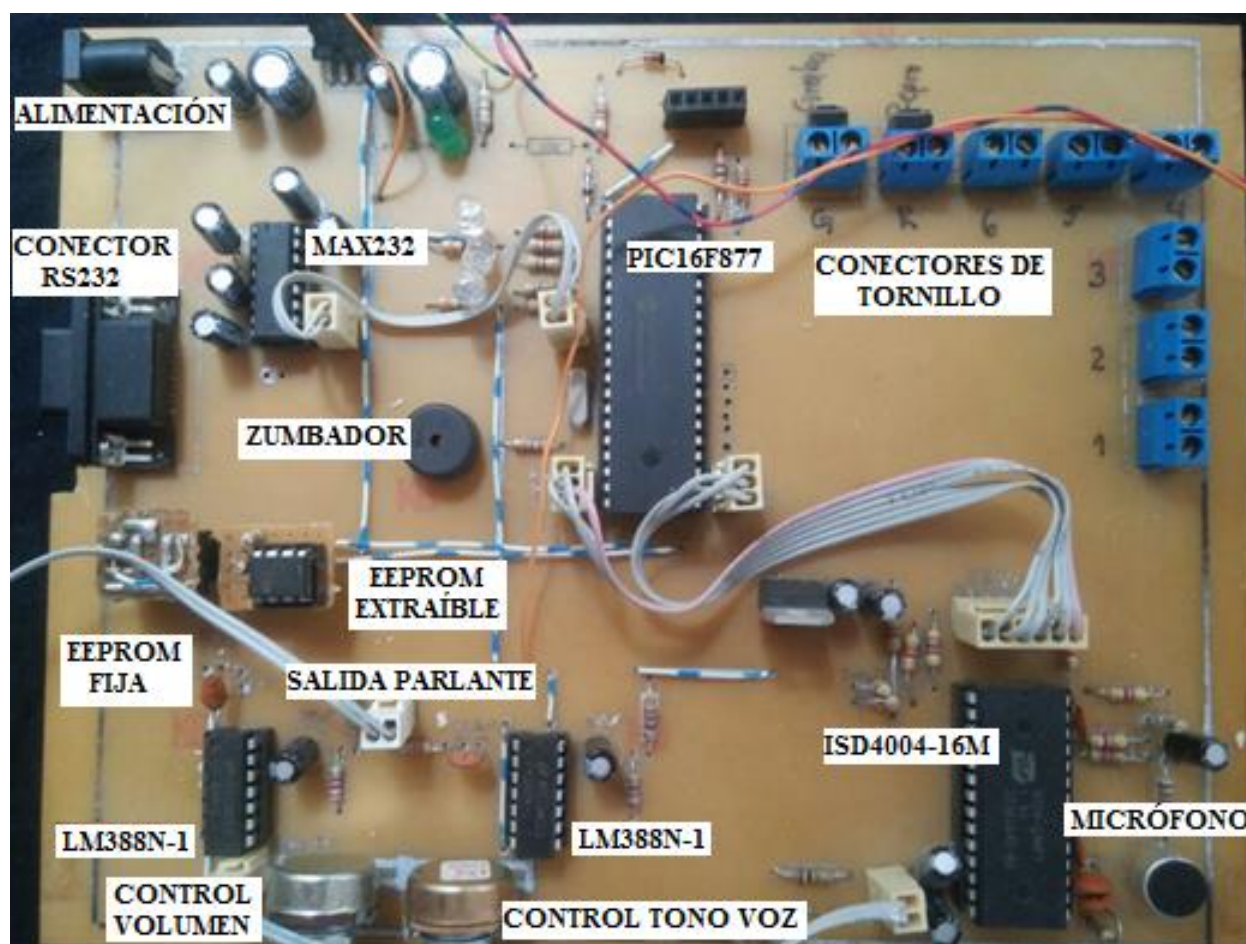


Ilustración 42: Tarjeta del Prototipo Hardware de un Apuntador Braille con Reproducción Silábica de Voz

El uso del Prototipo de un apuntador Braille con Reproducción Silábica de Voz, se puso a prueba con personas ciegas y con personas que no tienen esta limitación.



Ilustración 43: Usuario invidente escribiendo en Braille con el Prototipo



Ilustración 44: Usuario con visión normal escribiendo en Braille con el Prototipo

➤ **Resultados de las Etapas del Proyecto:**

❖ **Etapas de Entrada:** Durante el proceso de entrada de información de los datos al prototipo, se obtienen los siguientes resultados:

- Al momento de ingresar los apuntes por medio de la Celda Braille, el prototipo logra identificar todas las letras del abecedario español, los números del 0 al 1 y algunos caracteres. Dichos códigos se observan en las Tablas; 1,2 y 3. Para el caso específico de los números, se debe ingresar primero el código del carácter “#” y seguido el código del número deseado.

SÍMBOLO ASCII	CÓDIGO BRAILLE (RECIBE EL PIC) 654321	CÓDIGO CELDA BRAILLE
A	000001	1
B	000011	12
C	001001	14
D	011001	145
E	010001	15
F	001011	124
G	011011	1245
H	010011	125
I	001010	24
J	011010	245
K	000101	13
L	000111	123
M	001101	134
N	011101	1345
Ñ	111011	12456
O	010101	135
P	001111	1234
Q	011111	12345
R	011000	45
S	001110	234
T	011110	2345
U	100101	136
V	100111	1236
W	111010	2456
X	101101	1346
Y	111101	13456
Z	110101	1356

Tabla 1: Alfabeto Braille que reconoce el Prototipo

SÍMBOLO ASCII	CÓDIGO BRAILLE (RECIBE EL PIC) 654321	CÓDIGO CELDA BRAILLE
0	011010	245
1	000001	1
2	000011	12
3	001001	14
4	011001	145
5	010001	15
6	001011	124
7	011011	1245
8	010011	125
9	001010	24

Tabla 2: Números en Braille que reconoce el Prototipo

SÍMBOLO ASCII	CÓDIGO BRAILLE (RECIBE EL PIC) 654321	CÓDIGO CELDA BRAILLE
,	000010	2
.	000100	3
;	000110	23
:	010010	25
@	010000	5
*	010100	35
+	010111	1235
-	100100	36
/	001100	34
>	010110	235
<	101010	246
)	011100	345
?	100010	26
“	100110	236
=	110110	2356
#	111100	3456

Tabla 3: Caracteres en Braille que reconoce el Prototipo

Gracias a las pruebas de funcionamiento, se pudo notar que durante el proceso de la escritura en Braille, se debe esperar un lapso de tiempo (2 segundos

aproximadamente) para poder digitar letra por letra y que el prototipo las reconozca correctamente. Este resultado se convirtió en una limitante, ya que hace que el proceso de escritura en Braille con el Prototipo sea más extenso.

- En el proceso de grabación, se obtiene un buen comportamiento del ISD4004-16M, gracias a su variedad de frecuencias de muestreo que van desde los 4 hasta los 8KHz, lo cual permite elegir opciones de calidad de la voz.

- Se obtiene buen acople de todos los interruptores instalados, y buena respuesta de los pulsadores, ya que tienen alto grado de sensibilidad, los cuales facilitan el proceso de la escritura en Braille en el prototipo.

❖ **Etapas de Almacenamiento:** Se logra un buen resultado con respecto al proceso de almacenamiento en la memoria EEPROM Extraíble, ya que logra almacenar gran cantidad de información ingresada por el usuario al momento de tomar sus apuntes en Braille. Su capacidad máxima de almacenamiento es de treinta mil (30.000) caracteres, los cuales son precisos para que el usuario tome los apuntes completos en una clase.

❖ **Etapas de Procesamiento:** Con el PIC16F877A se tuvo éxito para poder cumplir con los objetivos del proyecto. Su memoria de programa fue utilizada en un 85% de su capacidad total. Gracias a su gran cantidad de pines, se deja habilitada la función de comunicación del prototipo con una computadora a través de un conector RS-232 instalado en el prototipo, para que estudiantes de la Universidad del Magdalena puedan usar este mismo proyecto como base para sus trabajos de grado, realizándole mejoras en su funcionamiento y hasta agregarle más funciones. Como el sistema Braille se basa en tan solo 6 botones que conforman el cajetín, este sistema se hace extenso para su adaptación total, y como la capacidad de memoria del cerebro del prototipo (PIC16F877A) es limitada, solo se logra adaptar los sonidos

para las sílabas que permiten formar palabras del idioma español y los sonidos para los números del cero (0) al nueve (9). Algunos caracteres son reconocidos por este prototipo pero no se logran adaptar sus pronunciaciones.

❖ **Etapas de Salida:** La implementación del ISD4004-16M fue satisfactoria y muy necesaria para cumplir con la etapa de la reproducción de este prototipo, objetivo muy importante en la realización de del proyecto. Gracias a su capacidad de grabación de voz, 16 minutos para esta referencia, se tiene el tiempo necesario para grabar las 942 sílabas que permiten formar palabras del idioma español.

Se logra instalar, adecuadamente, una etapa de amplificación de audio, configuración en puente de dos Amplificadores de Audio LM388N-1, la cual consta de 2 resistencias variables (potenciómetros) que permiten controlar el volumen y la tonalidad de la voz que se reproduce por medio del parlante.

Finalmente, se muestra el Prototipo completo, con todas sus etapas funcionando y listo para ser entregado:



Ilustración 45: Prototipo Final con sus partes funcionando

CAPÍTULO IV: CONCLUSIONES

11. CONCLUSIONES

El desarrollo del trabajo de grado cumplió con los objetivos propuestos inicialmente, ya que se diseñaron los esquemas y diagramas necesarios para fabricación del prototipo del hardware, el cual permite almacenar las palabras del lenguaje español utilizando el código Braille y reproducirlas silábicamente.

Al terminar este proyecto, se obtiene un prototipo de un Apuntador Braille con Reproducción Silábica de Voz con un Signo Generador Braille, en el cual, una persona con limitación visual que domine este sistema, está en toda la capacidad de escribir en él. Adicionalmente, con esta herramienta las personas que no tienen limitación visual pueden utilizarla para aprender a escribir en Braille, y enseñarle a una persona vidente, y así fomentar la inclusión de este sistema de aprendizaje en las instituciones educativas.

Al terminar las pruebas de uso del prototipo con personas invidentes y de visión normal, se obtienen críticas positivas por las 2 partes, como por ejemplos; fácil de usar, muy interactivo al momento de permitir personalizar el aparato con voz propia del usuario, muy práctico para transportar, entre otras.

Este proyecto abre las puertas a investigaciones futuras dentro del programa ingeniería electrónica de la Universidad del Magdalena, en donde no solo se tenga un prototipo que grabe, almacene y reproduzca sonidos para la enseñanza de personas con problemas de discapacidad visual, sino que además se puedan realizar técnicas para el aprendizaje. Adicionalmente, se puede manejar una base de datos de libros que puedan ser reproducidos en el dispositivo.

Se deja instalada una conexión RS-232 en el prototipo para que estudiantes de la Universidad del Magdalena puedan aprovechar este aparato y hacerle mejoras o tomarlo como base para sus futuros trabajos de Grado.

12. BIBLIOGRAFÍA

Mac Perez Lopez, 2011, El sistema braille, lectura para invidentes, recuperado de <http://www.discapacidadonline.com/sistema-braille-lectura-invidentes.html>

M.E.N. Ministerio de Educación Nacional, “estándares básicos de competencias del lenguaje”, p. 18, 05 de mayo del 2014. Consultado en Marzo 11 de 2016, disponible en http://www.mineducacion.gov.co/1621/articles-116042_archivo_pdf1.pdf

MÓNICA ISABEL HERNÁNDEZ RÍOS, publicación electronica sin fecha, LEY ESTATUTARIA 1618 DE 2013, recuperado de <http://discapacidadcolombia.com/index.php/legislacion/145-ley-estatutaria-1618-de-2013>

Dussán, María Antonieta, (2003), Dispositivos para limitados visuales desarrollados por el grupo aplicabilidad tecnologica de la UMB, Umbral Científico, núm. 3, pp. 66-73. Recuperado de <http://www.redalyc.org/articulo.oa?id=30400309>

Dussán Álvarez, María Antonieta; Jiménez Hernández, Luis Alexander; Hernández Suárez, Cesar Augusto; Giraldo Peñaranda, Leonel; Acosta Villamizar, Felipe, (2004), Sistema electrónico mecánico para el aprendizaje de la lecto-escritura del braille, Umbral Científico, núm. 5, pp. 59-65. Recuperado de <http://www.redalyc.org/pdf/304/30400309.pdf>

Cétares Salas Alfonso, Cortés Rivera Carlos, Silva Olarte Luis, (2005), Sistema De Enseñanza Del Código Braille Para Niños Con Limitaciones Visuales, (Proyecto de Pregrado). Universidad Javeriana, Bogotá - Colombia. Recuperado de <http://www.javeriana.edu.co/biblos/tesis/ingenieria/tesis85.pdf>

Hernández-Suarez Cesar y Jiménez-Hernández Luis, (2009), Desarrollo tecnológico para el mejoramiento de la comunicación a distancia entre personas con discapacidad, Rev. salud pública. 11 pp 828-835. Recuperado de <http://www.scielosp.org/pdf/rsap/v11n5/v11n5a16.pdf>

Balboa Lugo Miguel, Sánchez Benítez Félix, (2012), Entrenador Braille Electrónico Con Interfaz Usb De Hardware Y Software Abierto, (Proyecto de Pregrado), Instituto Tecnológico De Mérida, Yucatán – Mexico. Recuperado de <http://www.diee.net/wp-content/uploads/2014/11/Proyecto-Entrenador-Braille-Electr%C3%B3nico-con-interfaz-USB2.pdf>

Hernández-Suarez Cesar y Jiménez-Hernández Luis, (2011), Prototipo De Tecnología En Asistencia Para La Enseñanza Del Braille, Rev. Redes Ingenieria Vol 2 pp 105-115. Recuperado de <http://revistas.udistrital.edu.co/ojs/index.php/REDES/article/view/7168/8825>

Instituto De Tecnologías Educativas, “Educación inclusiva. Discapacidad visual”, No registra año de publicación, Recuperado de http://www.ite.educacion.es/formacion/materiales/129/cd/pdf/m2_dv.pdf

Edumatica, Voz Sintetica, 2008 recuperado de <http://edumatica.ing.ula.ve/politeca/Tecnoteca/Tenologicos/Edumatica/Fonomatica/Voz%20Sintetica/Instaladores/TextAloud/Voz%20Sint%20etica.pdf>

Microchip, PIC 16F877A, no registra años de publicación, recuperado de <http://www.microchip.com/wwwproducts/en/PIC16F877A>

López Vera, Julio Sebastián, (2005), Estudio analítico y experimental De los circuitos integrados de voz ISD, (Proyecto de Pregrado), Universidad Austral De Chile, Valdivia – Chile. Recuperado de <http://cybertesis.uach.cl/tesis/uach/2005/bmfci1864e/doc/bmfci1864e.pdf>

Cisneros Mejía María, (2010), Diseño E Implementación De Un Prototipo De Control De Tiempo Para Los Recorridos De Las Unidades De La Cooperativa De Transportes “Otavalo” Mediante Comunicación Inalámbrica En La Ruta Otavalo – Ibarra, (Proyecto de Pregrado), Universidad Técnica Del Norte, Ibarra – Ecuador. Recuperado de <http://repositorio.utn.edu.ec/bitstream/123456789/753/8/04%20RED%20003%20TESIS.pdf>

Booz, Allen & Hamilton Earned (2011). Consultado en Mayo 6, 2011. Value Management Tutorial Module 2: Work Breakdown Structure. Disponible en: http://science.energy.gov/~media/opa/powerpoint/Final_Module_2.ppt

NASA (2001). Consultado en Mayo 6, 2011. NASA procedimientos y directrices. Disponible en: <http://nodis3.gsfc>.

13. ANEXOS

➤ **Anexo 1: SÍLABAS QUE SE GRABAN EN EL ISD:** Son 942 sílabas en total, y se deben grabar en orden, de izquierda a derecha, empezando por el número 0, como se muestra a continuación:

0	1	2	3	4	5	6	7	8	9	a
ab	abs	ac	ad	ads	af	ag	al	am	an	ap
ar	as	at	ba	bab	bac	bad	bal	bam	ban	bap
bar	bas	be	vec	bed	bel	bem	ben	ber	bes	bi
vic	bid	bil	bin	bir	bis	bit	bla	blad	blan	blar
blas	ble	blen	bles	bli	blin	blis	blo	blon	blor	blor
blu	bo	vod	bol	bom	bon	bor	bos	bra	brad	bral
bran	brar	bras	bre	brel	bren	bres	bri	brid	bril	brin
brir	bris	bro	bron	bros	bru	brup	brus	bu	bul	bum
bun	bur	bus	ca	cac	cad	cal	cam	can	cap	car
cas	ce	sec	ced	seg	cel	sem	cen	cep	ceps	cer
ces	sex	cha	chad	chal	cham	chan	char	chas	che	chef
chen	ches	chi	chib	chid	chim	chin	chir	chis	cho	chom
chon	chor	chos	chu	chum	chun	chus	ci	sic	cid	sig
cil	cim	cin	cinc	cir	cis	cla	clan	clar	clas	cle
clec	clen	clep	cles	cli	clip	clis	clo	clon	clor	clos
clu	club	co	coc	cog	col	com	con	cons	cop	cor
cos	cot	cra	cral	cran	crar	cras	cre	cren	cres	cri
crin	crip	cris	cro	cron	cros	cru	crus	cu	cul	cum
cun	cuns	cur	cus	da	dac	dad	dal	dam	dan	dap
dar	das	de	ded	del	dem	den	dep	der	des	dex
di	dic	did	dif	dig	dil	dim	din	dip	dir	dis
dix	do	doc	dog	dol	dom	don	dop	dor	dos	dra
drac	drad	dral	dran	drar	dras	dre	dren	dres	dri	drid
dril	drin	drir	dris	dro	droc	dron	dros	dru	drus	du
duc	dul	dum	dun	dur	dus	e	ec	ed	eh	el
em	en	ep	er	es	et	ex	fa	fac	fal	fan
far	fas	fe	fec	fel	fen	fer	fes	fi	fic	fil
fin	fir	fis	flan	flar	flas	fle	flec	flen	fles	fli
flic	flir	flo	flon	flor	flu	fluc	fo	fol	fom	fon
for	fos	fra	frac	frag	fram	fran	frar	fras	fren	fren
fres	fri	fric	frin	frir	fro	fron	fru	fruc	frun	frus
fu	fuc	ful	fun	fur	fus	fut	ga	gad	gal	gam
gan	gar	gas	ge	jec	ged	gel	jem	gen	ger	ges
gi	gid	gil	gim	gin	gip	gir	gis	gla	glan	glar
glas	gle	gles	gli	glo	glon	glos	glu	go	gol	gon
gor	gos	got	gra	gral	gram	gran	grar	gras	gre	gren
gres	gri	grin	gris	gro	gros	gru	gu	gue	gui	guir
guis	gum	gun	gur	gus	i	ic	id	ig	il	im
in	ins	ip	ir	is	ist	ix	ja	jac	jad	jal
jam	jan	jar	jas	jo	jol	jon	jor	jos	ju	jun
jur	jus	la	lac	lad	lag	lal	lam	lan	lap	lar
las	lax	le	lec	lem	len	lep	ler	les	let	lex
li	lic	lid	lig	lim	lin	lip	lir	lis	lix	lla
llad	llan	llar	llas	lle	yec	yel	llen	ller	lles	lli
llin	llir	llis	llo	llon	llor	llos	llu	yun	yux	lo
loj	lom	lon	lop	lor	los	lu	luc	lud	lun	lup
lur	lus	ma	mad	mag	mal	mam	man	mar	mas	max

me	mec	med	mel	mem	men	mer	mes	mi	mic	mig
mil	mim	min	mip	mir	mis	mix	mo	mol	mon	mor
mos	mu	mud	mul	mum	mun	mur	mus	mut	na	nac
nad	nal	nam	nan	nap	nar	nas	ne	nec	ned	nel
nen	nep	ner	nes	nex	ni	nic	nid	nig	nil	nim
nin	nip	nir	nis	nit	nix	no	nob	noc	nol	nom
non	nop	nor	nos	nu	nul	num	nun	nup	nur	nus
ña	ñac	ñad	ñal	ñan	ñar	ñas	ñe	ñen	ñer	ñes
ñi	ñid	ñil	ñir	ñis	ño	ñol	ñon	ñor	ños	ñu
ñus	o	ob	obs	oc	of	og	ol	om	on	op
or	os	ov	ox	pa	pac	pad	pal	pam	pan	par
pas	pe	pec	ped	pel	pen	pep	per	pers	pes	pi
pic	pid	pig	pil	pim	pin	pir	pis	pla	plan	plar
plas	ple	plec	plen	ples	plex	pli	plin	plir	plis	plo
plon	plos	plot	plu	plum	plus	po	pol	pom	pon	pop
por	pos	post	pra	prac	prad	prag	pran	prar	pras	pre
preg	pren	pres	pri	prin	pris	pro	proc	prog	pron	pros
pru	pu	pug	pul	pun	pur	pus	que	quel	quen	quer
ques	quet	qui	quil	quin	quir	quis	ra	rac	rad	ral
ram	ran	rap	rar	ras	rax	re	red	reg	rel	rel
rem	ren	rep	rer	res	ret	ri	rid	ril	rim	rin
rir	ris	rit	rix	ro	rol	rom	ron	rop	ror	ros
rra	rral	rram	rran	rrar	rras	rre	rrec	rred	rreg	rrem
rren	rrer	rres	rri	rril	rrin	rrir	rris	rrit	rro	rrom
rron	rror	rros	rru	rrum	rrun	rrup	rrus	ru	ruc	rug
rum	run	rup	rur	rus	sa	sac	sad	sal	sam	san
sar	sas	so	sol	som	son	sor	sos	su	sub	suc
sud	sul	sun	sur	sus	ta	tac	tad	tal	tam	tan
tar	tas	tat	te	tec	ted	teg	tel	tem	ten	ter
tes	tex	ti	tic	tid	tig	til	tim	tin	tir	tis
tla	tlan	tlas	tlax	tle	tles	tlon	to	toc	tog	tol
tom	ton	top	tor	tos	tra	trac	trad	tral	tram	tran
trans	trar	tras	trax	tre	trel	tren	trep	trer	tres	tri
tric	tril	trin	trip	trir	tris	trol	trom	trom	tron	tros
tru	truc	trun	trus	tu	tud	tul	tum	tun	tur	tus
tut	u	ud	ul	um	un	ur	us	xa	xad	xal
xan	xar	xas	xe	xel	xen	xer	xes	xi	xil	xir
xis	xo	xon	xor	xos	xu	xul				

Ilustración 47: Sílabas que se graban en el ISD Parte 2

➤ **Anexo 2: CÓDIGO PARA PROGRAMAR EL PIC16F877A (Firmware):**

```

list          p=16f877a    ; list directive to define processor
#include       <p16f877a.inc>    ; processor specific variable definitions
#include       <Macrop16f877a.inc> ; macros

;=====
=====

; Registro para llevar la cuenta de cuántos mensajes se graban y de cuántos se reproducen
GLOBAL      RegAux1

; Registros utilizados para guardar cinco letras (una sílaba) proveniente de la
EEPROM_Sílaba
    global
    Letra1_EEPROM_Silaba,Letra2_EEPROM_Silaba,Letra3_EEPROM_Silaba,Letra4_EE
PROM_Silaba,Letra5_EEPROM_Silaba
    global w_temp

; Registro bandera para saber si el dato es un número o si es una letra
;    global Band_Ch
;    global TPaso,contapaso2
;    global PisoActual,PisoFuturo,Destinos

;    global UniCredi,DecCredi,CenCredi,UniMilCredi
;    global UniPremi,DecPremi,CenPremi,UniMilPremi
;    global Manzana,Sandia,Estrella,Siete,Bar,Campana,Melòn,Naranja,Cereza
;    global Cobrar,Bonos,Crèditos,Doblar_I,Doblar_D,Jugar
;    global BanderaSaldo

;=====
=====

```

; Rutinas

EXTERN

```
ret_1uSeg_20MHz,ret_2uSeg_20MHz,RET_10u_20MHZ,RET_100u_20MHZ
    extern RET_UDCMILI_20MHz,Ret_1Seg_20MHz,RET_1mSeg_20MHZ
EXTERN    BORRAM0,BORRAM1,BORRAM2,BORRAM3
extern LedOnOff
extern BuzzerOn
```

; Rutinas para manejar Eeprom Ext 24LC256

Extern

```
ConfigIIC,PrepaEscribaEeprom,EscribaEeprom,LeaEeprom,EscribaPWEeprom,Actua_D
ir_EEPROM,BorreEeprom
    Extern LeaEeprom_Silaba,Actua_Dir_EEPROM_Silaba
```

; Rutinas para la comunicación con el Pc

```
extern EnviarBytePc,RecibirBytePc
```

; Rutinas para manejar el ISD

```
; extern    Ini_Isd,isd_setrec,isd_rec,isd_stop,isd_setplay,isd_play
extern
```

```
GrabarISD_xSw,ReproducirISD_xSw,SaltaReproMensajeISD,SaltaLargo_ReproMensa_
ISD,PruebaSaltaLargoReproMensaISD,SaltaLargo1_ReproMensa_ISD
```

; Rutina principal de la descomposición silábica !!!

```
extern SeparaSilaba,Determ_Ch
```

; Rutina para conocer la dirección de la primera letra en la EEPROM_Sílaba en la cual
empezar la búsqueda por comparación

```
Extern BuscaAlfabetica
```

```

; Rutina para leer y dejar legible la sílaba contenida en la EEPROM_Sílaba
    extern LeaCargueEEPROMSilaba

; Rutina para pasar de Hexa a Decimal (queda guardado en W) el registro
"DatoRxDeEeprom"
    extern DeHexaADecEeprom_Silaba

; Otras rutinas
    extern
BorRegTrabLASPRGL,LeaActSumePRGuardeLetra,Dec_Dir_Eeprom,ActDirIniFinNota

; Rutina para determinar qué número de mensaje es la sílaba en el ISD, o sea cuál
mensaje se va a pronunciar
;    extern DetermNumMensaISD

; Rutinas para convertir de texto a voz
;    extern Determ_Ch

; Rutinas para las rutinas y subrutinas de descomposición silábica
;    extern DefineSiVocal

    EXTERN    TablaVD7SegmCC

; Tablas utilizadas en el proyecto
    extern TablaBraille_ASCII,TablaBraille_ASCII_Num
    extern TablaNumeros

;    extern
UnaSecuenciaSubir,UnaSecuenciaBajar,ArreLedsPiso,Visual7SegmPiso,Sonido
;    extern Actuar,DeterminePisoActual,DeterminePisoFuturo

```



```

;      extern CereTxByte  ; CereRxByte
;      extern CereTxPDU
;      extern MateCredi,LeaTeclado
;      extern Resta4Bytes,Suma4Bytes  ; Resta4bytes_2      ; Resta4Bytes

; Tablas
;      extern TablaDesp7Segm
;      extern TablaVD7SegmAscensor

; Registros

; Registros multipropósito
      extern ContaAux1,ContaAux2

; Registro para manejo de la comunicación con PC
      EXTERN      Viajero,BandFinMemEeprom

; Registros utilizados en el manejo del ISD
      extern DirBajaISD4004

; Registros utilizados para manejar la memoria Eeprom Externa 24LC256
      Extern
CtrlByteEscribirEe,DirAltaEeprom,DirBajaEeprom,CtrlByteLeerEe,DatoRxDeEeprom

; Registros utilizados para manejo de la EEPROM Sílabas 24LC256
      extern
CtrlByteEscribirEe_Silaba,CtrlByteLeerEe_Silaba,DirAltaEeprom_Silaba,DirBajaEeprom_Silaba
m_Silaba

```

```

; Registros utilizados para las rutinas y subrutinas de descomposición silábica
;extern Band_Vocal,ContaEspacio,
extern Band_Ch,PesoRelativoSílaba,Letra1,Letra2,Letra3,Letra4,Letra5

; Registros utilizados en la rutina de SaltoLargo_ReproMensaISD
extern
ContaAltoMensaISD,ContaBajoMensaISD,ContaAltoRutSaltoLargo,ContaBajoRutSalto
Largo

; Otros reg utilizados
extern DirAltaEeprom_L1,DirBajaEeprom_L1,ContaLetras

; Registros utilizados para guardar la ubicación de la sílaba en el ISD
; extern DirBajaSilabaISD,DirAltaSilabaISD

; EXTERN Viajero,banderasonido
; extern RespPORTD,ContaUnSeg_1,ContaUnSeg_2
; extern
UniM,UniS,DecM,DecS,CenM,CenS,UniMilM,UniMilS,UniR,DecR,CenR,UniMilR
; extern ContaMultiUsos
; extern DireBajaPagSorteo,DireAltaPagGuia,DireBajaPagGuia

```

```

MODULO_VAR    UDATA    0x20    ; Empieza Banco 0 (0x20 --> 0x6F) = 80
Registros Propósito General

```

```

; Variables utilizadas en las interrupciones
w_temp        RES    1        ; variable used for context saving
status_temp    RES    1        ; variable used for context saving
pclath_temp    RES    1        ; variable used for context saving

```

```

;ContaPrueba1      RES    1      ; Variable

BandNum    RES    1      ; Utilizada en la lectura Braille

; Variable para ISD
; TEMPOR    RES    1      ; Registro utilizada para guardar el número de veces que se
va a grabar/reproducir
RegAux1    RES    1

Letra1_EEPROM_Silaba  RES    1      ; Registro que guarda la letra 1 que se lee en
la EEPROM_Sílaba
Letra2_EEPROM_Silaba  RES    1      ; Registro que guarda la letra 2 que se lee en
la EEPROM_Sílaba
Letra3_EEPROM_Silaba  RES    1      ; Registro que guarda la letra 3 que se lee en
la EEPROM_Sílaba
Letra4_EEPROM_Silaba  RES    1      ; Registro que guarda la letra 4 que se lee en
la EEPROM_Sílaba
Letra5_EEPROM_Silaba  RES    1      ; Registro que guarda la letra 5 que se lee en
la EEPROM_Sílaba

; Variables en el banco 1
Var_FirmBanco1    UDATA    0xA0
Byte0 RES    1      ; 0xA0
Byte1 RES    1
Byte2 RES    1

```

Byte3	RES	1
Byte4	RES	1
Byte5	RES	1
Byte6	RES	1
Byte7	RES	1
Byte8	RES	1
Byte9	RES	1
Byte10	RES	1
Byte11	RES	1
Byte12	RES	1
Byte13	RES	1
Byte14	RES	1
Byte15	RES	1
Byte16	RES	1
Byte17	RES	1
Byte18	RES	1
Byte19	RES	1
Byte20	RES	1
Byte21	RES	1
Byte22	RES	1
Byte23	RES	1
Byte24	RES	1
Byte25	RES	1
Byte26	RES	1
Byte27	RES	1
Byte28	RES	1
Byte29	RES	1
Byte30	RES	1
Byte31	RES	1
Byte32	RES	1
Byte33	RES	1

Byte34	RES	1	
Byte35	RES	1	
Byte36	RES	1	
Byte37	RES	1	
Byte38	RES	1	
Byte39	RES	1	
Byte40	RES	1	
Byte41	RES	1	
Byte42	RES	1	
Byte43	RES	1	
Byte44	RES	1	
Byte45	RES	1	
Byte46	RES	1	
Byte47	RES	1	
Byte48	RES	1	
Byte49	RES	1	
Byte50	RES	1	
Byte51	RES	1	
Byte52	RES	1	
Byte53	RES	1	
Byte54	RES	1	
Byte55	RES	1	
Byte56	RES	1	
Byte57	RES	1	
Byte58	RES	1	
Byte59	RES	1	
Byte60	RES	1	
Byte61	RES	1	
Byte62	RES	1	
Byte63	RES	1	; 0xDF

```
,*****
```

```
RESET_VECTOR    CODE 0x000          ; processor reset vector
```

```
    nop                      ; nop required for icd
```

```
    movlw high start          ; load upper byte of 'start' label
```

```
    movwf PCLATH              ; initialize PCLATH
```

```
    goto start                ; go to beginning of program
```

```
start
```

```
    nop
```

```
    BANK0
```

```
    nop
```

```
    CLRF PORTA
```

```
    CLRF PORTB
```

```
    CLRF PORTC
```

```
    CLRF PORTD
```

```
    CLRF PORTE
```

```
    NOP
```

```
    BANK1
```

```
    NOP
```

```
    MOVLW    0x06    ; Puertos digitales
```

```
    MOVWF    ADCON1    ;
```

```
    NOP
```

```
    MOVLW    b'00100010'
```

```
    MOVWF    TRISA
```

```
    nop
```

```
    MOVLW    b'11111111'
```

```

MOVWF    TRISB ; ; Todas entradas para manejar el teclado Celda Braille

NOP

MOVLW    B'00000100' ; ; RC2 es una entrada para recibir señal RAC del
ISD

MOVWF    TRISC      ;

NOP

MOVLW    B'00000101' ; RD0 y RD2 son entradas para recibir señales del
ISD (INT y MISO respectivamente)

MOVWF    TRISD;

NOP

MOVLW    b'00000010' ; B'00000111' ; <RE0:RE3> entradas para
SW_Llamada <2:3>

MOVWF    TRISE

nop

pagesel    BORRAM0
call    BORRAM0


nop
BANK0
nop

;=====
=====
; Comienzo del programa

```

```

=====
;=====
=====

```

; 1. Configuro las palabras de control para escribir y para leer en las EEPROM

```

nop
movlw 0xA0 ; b'1010xxx0', donde xxx es la dirección del dispositivo en el bus !!!
movwf CtrlByteEscribirEe
nop

```

```

nop
movlw 0xA1 ; b'1010xxx1'... XXX corresponden a las direcciones en el bus

```

A2,A1 y A0

```

movwf CtrlByteLeerEe
nop

```

```

nop
movlw b'10101110' ; 0xA0 ; b'1010xxx0', donde xxx es la dirección del

```

dispositivo en el bus !!!

```

movwf CtrlByteEscribirEe_Silaba
nop

```

```

nop
movlw b'10101111' ; 0xA1 ; b'1010xxx1'... XXX corresponden a las direcciones

```

en el bus A2,A1 y A0

```

movwf CtrlByteLeerEe_Silaba
nop

```

; 3. Configuro las direcciones alta y baja de las Eeprom


```

nop
movlw 0x00 ; Primera dirección alta
movwf DirAltaEeprom
nop

```

```

nop
movlw 0x00 ; Primera dirección baja
movwf DirBajaEeprom
nop

```

```

nop
movlw 0x00 ; Primera dirección alta
movwf DirAltaEeprom_Silaba
nop

```

```

nop
movlw 0x00 ; Primera dirección baja
movwf DirBajaEeprom_Silaba
nop

```

; Prueba exactitud de la reproducción mensajes en la rutina salto entre mensajes

;PruebaRutSalto

```

;    movlw .1
;    movwf ContaAltoMensaISD
;    movlw .232
;    movwf ContaBajoMensaISD
;    nop
;    nop
;    nop

```

```
;    nop
```

; Como ya tengo las direcciones (alta y baja) en la que está el mensaje, procedo a saltar entre mensajes para llegar hasta la sílaba y pronunciarla...

```
;    pagesel    SaltaLargo1_ReproMensa_ISD    ;
SaltaLargo_ReproMensa_ISD
;    call    SaltaLargo1_ReproMensa_ISD    ; SaltaLargo_ReproMensa_ISD    ;
```

Va a buscar el mensaje entre todos los que hay en el ISD (Pronuncia la sílaba)

```
;    pagesel    LedOnOff
;    call    LedOnOff
```

```
;    pagesel    PruebaRutSalto
;    goto    PruebaRutSalto
```

Aqui

```
    nop
    pagesel    LedOnOff
    call    LedOnOff
```

```
    nop
    BANK0
    nop
```

```
    nop
    movlw b'01111111' ; 0xFF
    xorwf PORTB,W
    pagesel    VerificaNotas ; PregSws
```

```

    btfss STATUS,Z    ; Se pulsó la tecla RB7?
    goto VerificaNotas ; PregSws          ; No, así que revise Sw_Repro y
Sw_CxPc

    ;goto PregSws      ; Sí, ninguna tecla se ha digitado... Así que vuelva a revisar
SW_Repro y SW_CxPC y luego venga a leer el teclado Braille

```

```

;nop
;pagesel PregSws
;btfsc PORTB,7    ; ¿Se pulsó?
;goto PregSws     ; No

```

```

; Sí
nop
pagesel BorreEeprom
call BorreEeprom

```

Mientras

```

nop
pagesel LedOnOff
call LedOnOff
nop
pagesel Mientras
goto Mientras

```

; Bloque que se encarga de enumerar las notas según se vayan dando

VerificaNotas

```

nop
pagesel ActDirIniFinNota

```

call ActDirIniFinNota ; Rutina que se encarga de actualizar las direcciones de inicio y de fin de nota...

; Se utiliza para la gestión de la EepromNotas (para posteriormente poder enumerar cada Nota)

; Se pregunta si ambos suiches están activos a la vez...

PregSws

nop

pagesel PregSw_Repro

btfss Sw_Repro ; ¿En 5V?...RA1

goto PregSw_Repro ; No

; Sí

nop

nop

nop

nop

nop

pagesel PregSw_Repro

btfss Sw_CxPc ; ¿En 5V?... RA5

goto PregSw_Repro ; No

; Sí

nop

nop

nop

nop

; Como se da una condición de exceso de consumo de corriente (Se alimenta a la vez el MAX y se alimenta ISD-AMP), se activa un buzzer indicando que ambos SW están ON

nop

```

pagesel    BuzzerOn
call       BuzzerOn
nop
pagesel    PregSws
goto       PregSws    ; Aqui

```

PregSw_Repro

```

nop
nop
nop
nop
nop
pagesel    PregSw_CxPc
btfss     Sw_Repro    ; ¿Está en 1?
goto       PregSw_CxPc    ; No
; Sí
bsf        Led1
nop
nop
nop
nop
nop
nop
bcf        Led2
nop
pagesel    Desarr_ISD    ; PregSws
goto       Desarr_ISD    ; PregSws    ; Aqui

```

PregSw_CxPc

```

nop
nop
nop

```

```

    nop
    nop
    pagesel    LeeCeldaBraille
    btfss    Sw_CxPc    ; ¿Está en 1?
    goto    LeeCeldaBraille    ; No
    ; Sí
    bsf    Led2
    nop
    nop
    nop
    nop
    nop
    bcf    Led1

;    nop
;    pagesel    IniLeaEeprom_TxPc
;    goto    IniLeaEeprom_TxPc

```

; Primero, el micro espera a que el PC le envíe la orden a desarrollar : puede ser que reciba datos y los almacene en EEPROM o puede ser que lea EEPROM y se los envíe...

RxPc_Asterisco

```

    nop
    pagesel    RecibirBytePc
    call    RecibirBytePc; Recibe un byte y lo guarda en reg. Viajero

;    nop
;    pagesel    EnviarBytePc
;    call    EnviarBytePc
;May0905
;    nop
;    pagesel    LedOnOff

```

```

;    call    LedOnOff

;    nop

;    pagesel    RxPc_Asterisco    ; May0905
;    goto    RxPc_Asterisco    ; May0905

    nop
    nop
    nop
    nop
    movlw "*"
    xorwf    Viajero,W
    pagesel    RxPc_Asterisco
    btfss    STATUS,Z    ; ¿Es el asterisco?
    goto    RxPc_Asterisco    ; No, así que se queda esperándolo en un ciclo
cerrado...

;    movlw    "O"
;    movwf    Viajero
;    pagesel    EnviarBytePc
;    call    EnviarBytePc

;    Sí
;    Ahora se recibe el segundo byte que contiene la orden a ejecutar
    nop
    pagesel    RecibirBytePc
    call    RecibirBytePc; Recibe un byte

;    Se pregunta si el PC quiere leer "L" la EEPROM

```

RxPc_L

```

    nop
    movlw "L"
    xorwf Viajero,W
    pagesel    RxPc_Asterisco    ;
    btfss  STATUS,Z    ; ¿Es la orden de leer EEPROM?
    goto  RxPc_Asterisco    ;    ; No

    ; Sí
    nop
    pagesel    IniLeaEeprom_TxPc
    goto  IniLeaEeprom_TxPc

```

; Bloque para la transmisión de los datos EEPROM al PC

IniLeaEeprom_TxPc

```

    nop
    BANK0    ; Me aseguro de estar en el banco 0
    nop

    nop
    bcf    Led
    nop
    nop
    nop
    nop
    nop
    nop
    bsf    Led1

    nop

```



```

movlw "O"
movwf Viajero
pagesel    EnviarBytePc
call      EnviarBytePc

```

```

nop
movlw "K"
movwf Viajero
pagesel    EnviarBytePc
call      EnviarBytePc

```

;----- Anex0_12082014... Direcciono la Eeprom para que envíe las NOTAS desde la posición .1000 al PC

```

Anex0_12082014
    movlw b'00000011'
    movwf DirAltaEeprom
    movwf b'11101000'
    movwf DirBajaEeprom
    nop

```

Sale_Anex0_12082014

```

    nop

```

;----- Anex0_12082014... Direcciono la Eeprom para que envíe las NOTAS una a una al PC

LeaEeprom_TxPc

```

    nop
    pagesel    LeaEeprom
    call      LeaEeprom

```

```

    nop

```

BANK0 ; Me aseguro de estar en el banco 0

nop

movfw DatoRxDeEeprom ; Tomo el dato leído de la EEPROM

movwf Viajero ; y lo pongo en el reg. Viajero

nop

pagesel EnviarBytePc

call EnviarBytePc ; para transmitirlo al Pc

nop

pagesel Actua_Dir_EEPROM

call Actua_Dir_EEPROM; Actualiza en el PIC la dirección en la que va
escritura de la EEPROM

nop

movlw 0xff

xorwf BandFinMemEeprom,W

pagesel LeaEeprom_TxPc

btfss STATUS,Z ; ¿Iguales?... Se llegó al final de la memoria?

goto LeaEeprom_TxPc ; No

; Sí, fin de memoria

movlw "F"

movwf Viajero

pagesel EnviarBytePc

call EnviarBytePc

movlw "I"

movwf Viajero

pagesel EnviarBytePc

```
call    EnviarBytePc
```

```
movlw "N"
```

```
movwf Viajero
```

```
pagesel    EnviarBytePc
```

```
call    EnviarBytePc
```

FinTxPcMemEeprom

```
nop
```

```
pagesel    BuzzerOn
```

```
call    BuzzerOn
```

```
nop
```

```
pagesel    FinTxPcMemEeprom
```

```
goto    FinTxPcMemEeprom ; Ciclo cerrado indicando que ya se terminó la
```

operación

LeeCeldaBraille

; Se pregunta si alguna tecla fué deflectedada

```
nop
```

```
movlw 0xFF
```

```
xorwf PORTB,W
```

```
pagesel    Repite_LeeCeldaBraille
```

```
btfss    STATUS,Z    ; Teclado sin pulsar?
```

```
goto    Repite_LeeCeldaBraille    ; PregSws    ; No, alguna tecla se
```

pulsó, así que convierta de Braille a ASCII

```
nop
```

```
pagesel    PregSws
```

```
goto    PregSws    ; Sí, ninguna tecla se ha digitado... Así que vuelva a revisar
```

SW_Repro y SW_CxPC y luego venga a leer el teclado Braille

Repite_LeeCeldaBraille

```

    nop
    movlw .200
    pagesel    RET_UDCMILI_20MHz
    call    RET_UDCMILI_20MHz    ; Retardo antirebote

```

```

    nop
    movlw 0xFF
    xorwf PORTB,W
    pagesel    DefinaSimboloBraille
    btfss    STATUS,Z    ; Teclado sin pulsar?
    goto    DefinaSimboloBraille; PregSws    ; No, alguna tecla se pulsó, así

```

que convierta de Braille a ASCII

```

    ; Sí
    nop
    pagesel    PregSws    ;    Aquí
    goto    PregSws    ;    Aquí ; PregSws    ; Ninguna tecla se ha

```

digitado... Así que vuelva a revisar SW_Repro y SW_CxPC y luego venga a leer el teclado Braille

DefinaSimboloBraille

```

    nop
    movfw PORTB
    movwf Viajero
    nop
    nop
    nop

```

; Caso en el que digitan el símbolo de número

```

nop
movlw .195    ; b'11000011' como lo ve el teclado, antes de hacer el complemento
xorwf Viajero,W
pagesel      SigaDefinaSimboloBraille
btfss STATUS,Z    ; ¿Iguales?
goto SigaDefinaSimboloBraille ; No

```

```

; Sí
nop
movlw 0xFF
movwf BandNum    ; Bandera puesta

```

```

nop
PAGESEL LeeCeldaBraille
GOTO LeeCeldaBraille ; Regrese a esperar el número que viene

```

SigaDefinaSimboloBraille

```

nop
movlw 0xFF
xorwf BandNum,W
pagesel      DefinaNumBraille
btfsc STATUS,Z    ; ¿Iguales?
goto DefinaNumBraille ; Sí
; No

```

```

nop
nop
nop
nop
comf Viajero,1
nop

```

nop

nop

nop

nop

nop

movlw .64

subwf Viajero,W

pagesel LeeCeldaBraille

btfsc STATUS,C ; ¿Igual o mayor?

GOTO LeeCeldaBraille ; Sí, es mayor... Vuelva a leer la celda Braille porque
está fuera del rango de la tabla

; No, es menor

nop

movfw Viajero

nop

pagesel TablaBraille_ASCII

call TablaBraille_ASCII

nop

nop

movwf Viajero

nop

nop

nop

movlw .40 ; quieren: teclas mayúsculas... Pero como el sistema trabaja con
mayúsculas, entonces se obvia esta orden...

xorwf Viajero,W

pagesel LeeCeldaBraille ; Regresa a leer la ste. tecla

```

btfsc STATUS,Z ; ¿Iguales?
goto LeeCeldaBraille ; Sí... ; Regresa a leer la ste. tecla

```

```

; No

```

```

; ----- Anex0_09082014... Intento enumerar cada nota en el índice y escribir cada
nota en su respectiva página

```

```

; Entonces pregunta si es la primera vez que pasa por aquí

```

```

Anex0_09082014

```

```

nop
movlw 0xff
xorwf Letra5,W
pagesel SalgaAnex0_09082014
btfss STATUS,Z ; ¿Iguales?
goto SalgaAnex0_09082014 ; No

```

```

; Sí

```

```

clrf Letra5 ; Borro la bandera para no volver a entrar en este anexo

```

```

movlw b'00000011'
movwf DirAltaEeprom
movlw b'11011110'
movwf DirBajaEeprom ; Dir .990 (En el índice, donde está el conteo global

```

```

de las Notas)

```

```

pagesel LeaEeprom
call LeaEeprom
movfw DatoRxDeEeprom
movwf ContaLetras ; Se guarda el valor de las decenas del conteo

```

```

pagesel Actua_Dir_EEPROM

```

call Actua_Dir_EEPROM; Se apunta hacia las unidades del conteo

pagesel LeaEeprom

call LeaEeprom

movfw DatoRxDeEeprom

movwf PesoRelativoSílaba ; Se guarda el valor de las unidades del conteo

; Se pregunta si el conteo global llegó a .99

movlw .9

xorwf ContaLetras,W

pagesel Ste_Anex0_09082014

btfss STATUS,Z ; Igual?

goto Ste_Anex0_09082014 ; No

; Sí

pagesel BuzzerOn

call BuzzerOn ; Aviso que la memoria está al 90% o más de su capacidad

movlw .9

xorwf PesoRelativoSílaba,W

pagesel Ste_Anex0_09082014

btfss STATUS,Z ; Igual?

goto Ste_Anex0_09082014 ; No

; Sí, entonces la memoria llegó al final de su capacidad y no puede seguir almacenando apuntes, debe reemplazarse o formatearse...

ErrorMemNotasLlena

nop

pagesel BuzzerOn

call BuzzerOn

nop


```

pagesel    ErrorMemNotasLlena
goto       ErrorMemNotasLlena

```

Ste_Anex0_09082014

; Se incrementa el valor del conteo

```

movlw .9
xorwf    PesoRelativoSílaba,W
pagesel    Ajuste1Anex0_09082014
btfss    STATUS,Z    ; Igual?
goto     Ajuste1Anex0_09082014    ; No

```

; Sí

```

incf     ContaLetras,1
clrf     PesoRelativoSílaba
pagesel    EscribaDirAnex0_09082014
goto     EscribaDirAnex0_09082014

```

Ajuste1Anex0_09082014

```

incf     PesoRelativoSílaba,1

```

EscribaDirAnex0_09082014

; Escribo la nueva dirección en el conteo global (índice .990)

```

nop
nop
nop
nop
nop
movlw b'00000011'
movwf DirAltaEeprom
movlw b'11011110'

```

movwf DirBajaEeprom ; Dir .990 (En el índice, donde está el conteo global de las Notas)

```

pagesel    PrepaEscribaEeprom
call       PrepaEscribaEeprom
movfw     ContaLetras
pagesel    EscribaEeprom
call       EscribaEeprom ; Escribo las decenas del conteo global

```

```

pagesel    Actua_Dir_EEPROM
call       Actua_Dir_EEPROM; Dir .991

```

```

pagesel    PrepaEscribaEeprom
call       PrepaEscribaEeprom
movfw     PesoRelativoSílabas
pagesel    EscribaEeprom
call       EscribaEeprom ; Escribo las unidades del conteo global

```

; Ahora escribo, en el índice, el número de la Nota en el lugar correspondiente a dicha nota

; Primero cargo la dirección de la nota correspondiente (sigo en el índice)

```

movfw     Letra3
movwf     DirAltaEeprom
movfw     Letra4

```

movwf DirBajaEeprom ; Apunto hacia las decenas del conteo en el reglón de la nota en cuestión (en el índice)

```

pagesel    Dec_Dir_Eeprom
call       Dec_Dir_Eeprom ; Apunto hacia las unidades del conteo (en el reglón de la nota correspondiente)

```

```

pagesel    PrepaEscribaEeprom

```

```

call    PrepaEscribaEeprom
movfw  ContaLetras
pagesel    EscribaEeprom
call    EscribaEeprom    ; Escribo las decenas del conteo en la Nota
correspondiente (sigo en el índice-Renglón)

```

```

pagesel    Actua_Dir_EEPROM
call    Actua_Dir_EEPROM;

```

```

pagesel    PrepaEscribaEeprom
call    PrepaEscribaEeprom
movfw  PesoRelativoSílaba
pagesel    EscribaEeprom
call    EscribaEeprom    ; Escribo las inidades del conteo en la Nota
correspondiente (sigo en el índice-Renglón)

```

```

pagesel    Actua_Dir_EEPROM
call    Actua_Dir_EEPROM;

```

; Ahora se lee la dirIni de la Nota en la que voy a escribir, lo hago del renglón global

```

pagesel    LeaEeprom
call    LeaEeprom
movfw  DatoRxDeEeprom
movwf  ContaAux1    ; Leo y guardo en RAM la dirAlta de dónde empieza la
Nota

```

```

pagesel    Actua_Dir_EEPROM
call    Actua_Dir_EEPROM; Dir .993

```

```

pagesel    LeaEeprom
call    LeaEeprom

```

```
movfw DatoRxDeEeprom
```

```
movwf ContaAux2 ; Leo y guardo en RAM la dirBaja de dónde empieza la
```

Nota

```
; Se pregunta si se llegó al máximo de direcciones permitido en la EepromNotas
```

```
movlw b'01111011'
```

```
subwf ContaAux1,W
```

```
pagesel ErrorMemNotasLlena
```

```
btfsc STATUS,C ; Mayor o igual de 31488 (en Dir)?
```

```
goto ErrorMemNotasLlena; Sí
```

```
; No
```

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```
; Se pregunta si se llegó al 90% o más de la capacidad de la EepromNotas
```

```
movlw b'01101101'
```

```
subwf ContaAux1,W
```

```
pagesel BuzzerOn
```

```
btfsc STATUS,C ; Mayor o igual de 27904 (en Dir)?
```

```
call BuzzerOn ; Sí
```

```
; No
```

; Ahora se va a escribir "NOTA " en la EepromNotas al comienzo de esta Nota (No en el índice)

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```

movfw ContaAux1
movwf DirAltaEeprom
movfw ContaAux2
movwf DirBajaEeprom

```

```

pagesel      PrepaEscribaEeprom
call  PrepaEscribaEeprom
movlw "N"
pagesel      EscribaEeprom
call  EscribaEeprom      ; Escribo "N" al inicio de la Nota

```

```

pagesel      Actua_Dir_EEPROM
call  Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

```

pagesel      PrepaEscribaEeprom
call  PrepaEscribaEeprom
movlw "O"
pagesel      EscribaEeprom
call  EscribaEeprom      ; Escribo "O" al inicio de la Nota

```

```

pagesel      Actua_Dir_EEPROM
call  Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

```

pagesel      PrepaEscribaEeprom
call  PrepaEscribaEeprom
movlw "T"
pagesel      EscribaEeprom
call  EscribaEeprom      ; Escribo "N" al inicio de la Nota

```

```

pagesel    Actua_Dir_EEPROM
call    Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

```

pagesel    PrepaEscribaEeprom
call    PrepaEscribaEeprom
movlw "A"
pagesel    EscribaEeprom
call    EscribaEeprom      ; Escribo "N" al inicio de la Nota

```

```

pagesel    Actua_Dir_EEPROM
call    Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

```

pagesel    PrepaEscribaEeprom
call    PrepaEscribaEeprom
movlw 0x20
pagesel    EscribaEeprom
call    EscribaEeprom      ; Escribo espacio al inicio de la Nota

```

```

pagesel    Actua_Dir_EEPROM
call    Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

; Ahora escribo el número de la Nota

; pero primero debo preguntar si las decenas están en ceros

```

movlw .0
xorwf  ContaLetras,W
pagesel    EscribDec_Num
btfss  STATUS,Z    ; ¿Iguales?
goto    EscribDec_Num    ; No

```

; Sí

EscRibDec_BarraPiso

nop

pagesel PrepaEscribaEeprom

call PrepaEscribaEeprom

movlw 0x5f ; Símbolo "_"

pagesel EscribaEeprom

call EscribaEeprom ; Escribo las decenas del conteo al inicio de la

págNota correspondiente

pagesel EscribUni_Num

goto EscribUni_Num

EscRibDec_Num

pagesel PrepaEscribaEeprom

call PrepaEscribaEeprom

movfw ContaLetras

pagesel EscribaEeprom

call EscribaEeprom ; Escribo las decenas del conteo al inicio de la

págNota correspondiente

EscRibUni_Num

pagesel Actua_Dir_EEPROM

call Actua_Dir_EEPROM; Apunto hacia la ste dirección en la PágNota

correspondiente

pagesel PrepaEscribaEeprom

call PrepaEscribaEeprom

movfw PesoRelativoSílabas

pagesel EscribaEeprom

call EscribaEeprom ; Escribo las unidades del conteo al inicio de la

págNota correspondiente

```

    pagesel      Actua_Dir_EEPROM
    call  Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

```

    pagesel      PrepaEscribaEeprom
    call  PrepaEscribaEeprom
    movlw 0x20
    pagesel      EscribaEeprom
    call  EscribaEeprom      ; Escribo espacio al inicio de la Nota

```

```

    pagesel      Actua_Dir_EEPROM
    call  Actua_Dir_EEPROM; Apunto hacia la ste dirección para la próxima
escritura (escribo en la págNota)

```

; Ahora ya puedo salir tranquilamente

```

SalgaAnex0_09082014
; ----- Fin Anex0_09082014... Intento enumerar cada nota en el índice y escribir
cada nota en su respectiva página

```

```

    nop
    pagesel      PrepaEscribaEeprom
    call  PrepaEscribaEeprom
    nop
    movfw Viajero

```

```

    nop
    pagesel      EscribaEeprom
    call  EscribaEeprom

```



```

        nop
        pagesel    Actua_Dir_EEPROM
        call    Actua_Dir_EEPROM; Actualiza en el PIC la dirección en la que va
escritura de la EEPROM

```

```

;        nop
;        movlw .200
;        pagesel    RET_UDCMILI_20MHz
;        call    RET_UDCMILI_20MHz    ; Retardo esperando que el usuario haya
soltado los botones Braille

```

```

        nop
        pagesel    LedOnOff
        call    LedOnOff    ; Para visualizar cuándo se puede volver a anotar

```

```

        nop
        PAGESEL    LeeCeldaBraille
        GOTO LeeCeldaBraille ;no clear next

```

```

;-----
-----
; Variante para definir un número Braille
;-----
-----

```

DefinaNumBraille

```

        nop
        clrf    BandNum    ; Se borra la bandera
        nop

```

nop

nop

nop

nop

comf Viajero,1

nop

nop

nop

nop

movlw .28

subwf Viajero,W

pagesel LeeCeldaBraille

btfsc STATUS,C ; ¿Igual o mayor?

GOTO LeeCeldaBraille ; Sí, es mayor... Vuelva a leer la celda Braille porque
está fuera del rango de la tabla

; No, es menor
nop

nop

movfw Viajero

nop

pagesel TablaBraille_ASCII_Num

call TablaBraille_ASCII_Num

nop

nop

movwf Viajero

nop

pagesel PrepaEscribaEeprom

```
call    PrepaEscribaEeprom
```

```
nop
```

```
movfw Viajero
```

```
nop
```

```
pagesel    EscribaEeprom
```

```
call    EscribaEeprom
```

```
nop
```

```
pagesel    Actua_Dir_EEPROM
```

```
call    Actua_Dir_EEPROM; Actualiza en el PIC la dirección en la que va
escritura de la EEPROM
```

```
;    nop
```

```
;    movlw .200
```

```
;    pagesel    RET_UDCMILI_20MHz
```

```
;    call    RET_UDCMILI_20MHz    ; Retardo esperando que el usuario haya
soltado los botones Braille
```

```
nop
```

```
pagesel    LedOnOff
```

```
call    LedOnOff    ; Para visualizar cuándo se puede volver a anotar
```

```
;    incf    FSR,F
```

```
;    movlw 0xE0
```

```
;    XORWF    FSR,W
```

```
;    PAGESEL    LeeCeldaBraille
```

```
;    BTFSS STATUS,Z ;all done?
```

```
; GOTO LeeCeldaBraille ;no clear next
; RETLW 0
```

```
nop
PAGESEL LeeCeldaBraille
GOTO LeeCeldaBraille ;no clear next
```

```
;=====
=====
; Trabajo con el ISD
;=====
=====
```

```
Desarr_ISD
; Aquí pregunta si se defleca la tecla correspondiente para grabar voz
nop
pagesel GrabandoMensajesISD_xSw
btfsc Sw_ReproMensaje ; Se pulsó RB6?
goto GrabandoMensajesISD_xSw ; No... Entonces mire si se quiere grabar
mensajes...
; Sí
nop
movlw .100
pagesel RET_UDCMILI_20MHz
call RET_UDCMILI_20MHz ; Retardo antirebote
```

```
; Verificación de pulso válido para reproducir un mensaje...
```

```
nop
pagesel GrabandoMensajesISD_xSw
btfsc Sw_ReproMensaje ; Se pulsó RB6?
```

goto GrabandoMensajesISD_xSw ; No... Entonces mire si se quiere grabar
mensajes...

nop
pagesel ReproMensajesISD_xSw
goto ReproMensajesISD_xSw

GrabandoMensajesISD_xSw

; Aquí pregunta si se defleca la tecla correspondiente para grabar voz

nop
pagesel Desarr_ISD
btfsc Sw_GrabaMensaje ; Se pulsó RB7?
goto Desarr_ISD ; No... Siga esperando en un ciclo cerrado a que se graben
mensajes

; Sí

nop
movlw .100
pagesel RET_UDCMILI_20MHz
call RET_UDCMILI_20MHz ; Retardo antirebote
; Verificación de pulso válido...

nop
pagesel Desarr_ISD
btfsc Sw_GrabaMensaje ; Se pulsó RB7?
goto Desarr_ISD ; No... Siga esperando en un ciclo cerrado a que se graben
mensajes

; Sí, el pulso es válido

; Se graba el mensaje... Empieza en dirección(alta y baja) = 0x00

nop

```
pagesel    GrabarISD_xSw
```

```
call    GrabarISD_xSw    ; Graba todos los mensajes que se quieran/puedan...
```

OJOOOOO: esta rutina es un ciclo cerrado!!!

```
;-----
```

```
;*****
```

```
*****
```

```
*****
```

```
; El siguiente bloque se encarga de la reproducción sonora de lo que esté grabado en la
memoria
```

```
;*****
```

```
*****
```

```
*****
```

```
;-----
```

```
; 18Jun2014... Sin este bloque el espacio en memoria de programa es de 1049
```

```
ReproMensajesISD_xSw
```

```
; Bloque que se encarga de hacer la separación silábica de una palabra
```

```
    nop
```

```
;    ; Me aseguro de que la primera vez se lea la Eeprom, se dirija a la dirección 0x00
```

```
;    clrf    DirAltaEeprom
```

```
;    clrf    DirBajaEeprom
```

```
; La dirección inicial es al comienzo de la Nota1
```

```
    movlw 0x03
```

```
    movwf DirAltaEeprom
```

```
movlw 0xe8
movwf DirBajaEeprom
```

Ste_ReproMensajesISD_xSw

```
pagesel      SeparaSilaba
call  SeparaSilaba ; Las sílabas salen con un peso relativo con el cual se busca
su mensajeISD,
; Los números vienen directamente
con su direcciónISD
; Los signos ortográficos no necesitan
pronunciarse...
```

```
; Se pregunta si es un caracter inválido
Pronuncia_Ch_Invalido
;      movlw 0x00
;      xorwf Band_Ch,W
;      pagesel      Ste_ReproMensajesISD_xSw      ;
AjusteReproMensajesISD_xSw
;      btfsc STATUS,Z ; Es un carater inválido?
;      goto  Ste_ReproMensajesISD_xSw      ; AjusteReproMensajesISD_xSw      ;
Sí
;      ; No
```

; ----- Le hago este anexo pensando en pronunciar el No. del conteo
(viene en conteo decimal y necesito pasarlo a ASCII

Anex1_11082014

```

movlw 0x00
xorwf Band_Ch,W
pagesel Pronuncia_Numero ; Ste_ReproMensajesISD_xSw ;
AjusteReproMensajesISD_xSw
btfss STATUS,Z ; Es un carater inválido?
goto Pronuncia_Numero ; Ste_ReproMensajesISD_xSw ;
AjusteReproMensajesISD_xSw ; No
; Sí

```

```

movlw .0
xorwf DatoRxDeEeprom,W
pagesel Ste_ReproMensajesISD_xSw
btfsc STATUS,Z ; ¿Es cero?
goto Ste_ReproMensajesISD_xSw ; Sí
; No

```

```

movlw .1
xorwf DatoRxDeEeprom,W
pagesel Ajuste_Anex1_11082014
btfsc STATUS,Z ; ¿Es uno?
goto Ajuste_Anex1_11082014 ; Sí
; No

```

```

movlw .2
xorwf DatoRxDeEeprom,W
pagesel Ajuste_Anex1_11082014
btfsc STATUS,Z ; ¿Es uno?
goto Ajuste_Anex1_11082014 ; Sí
; No

```

```

movlw .3

```



```

xorwf DatoRxDeEeprom,W
pagesel      Ajuste_Anex1_11082014
btfsc STATUS,Z    ; ¿Es uno?
goto  Ajuste_Anex1_11082014    ; Sí
; No

```

```

movlw .4
xorwf DatoRxDeEeprom,W
pagesel      Ajuste_Anex1_11082014
btfsc STATUS,Z    ; ¿Es uno?
goto  Ajuste_Anex1_11082014    ; Sí
; No

```

```

movlw .5
xorwf DatoRxDeEeprom,W
pagesel      Ajuste_Anex1_11082014
btfsc STATUS,Z    ; ¿Es uno?
goto  Ajuste_Anex1_11082014    ; Sí
; No

```

```

movlw .6
xorwf DatoRxDeEeprom,W
pagesel      Ajuste_Anex1_11082014
btfsc STATUS,Z    ; ¿Es uno?
goto  Ajuste_Anex1_11082014    ; Sí
; No

```

```

movlw .7
xorwf DatoRxDeEeprom,W
pagesel      Ajuste_Anex1_11082014
btfsc STATUS,Z    ; ¿Es uno?

```

```
goto Ajuste_Anex1_11082014 ; Sí
; No
```

```
movlw .8
xorwf DatoRxDeEeprom,W
pagesel Ajuste_Anex1_11082014
btfsc STATUS,Z ; ¿Es uno?
goto Ajuste_Anex1_11082014 ; Sí
; No
```

```
movlw .9
xorwf DatoRxDeEeprom,W
pagesel Ajuste_Anex1_11082014
btfsc STATUS,Z ; ¿Es uno?
goto Ajuste_Anex1_11082014 ; Sí
; No
```

```
pagesel Ste_ReproMensajesISD_xSw
goto Ste_ReproMensajesISD_xSw
```

Ajuste_Anex1_11082014

```
nop
incf DatoRxDeEeprom,1
nop
nop
nop
nop
nop
nop
pagesel BuscaMensaISD_Num
goto BuscaMensaISD_Num
```

```

;    movlw .10
;    subwf DatoRxDeEeprom,W
;    pagesel    Ste_ReproMensajesISD_xSw    ; Pronuncia_Numero
;    btfsc  STATUS,C    ; ¿Mayor o igual?
;    goto  Ste_ReproMensajesISD_xSw    ; Pronuncia_Numero ;Si
;    ; No

```

```

;    movlw 0x30
;    addwf DatoRxDeEeprom,1
;    nop
;    nop
;    nop
;    movlw 0x10
;    movwf Band_Ch
;    nop
;    pagesel    BuscaMensaISD_Num
;    goto  BuscaMensaISD_Num

```

; ----- Fin... Le hago este anexo pensando en pronunciar el No. del
conteo (viene en conteo decimal y necesito pasarlo a ASCII)

; Se pregunta si es un número

Pronuncia_Numero

```

    nop
    nop
    nop
    nop
    nop
    movlw 0x10
    xorwf Band_Ch,W
    pagesel    BuscaMensaISD_Num

```

```

    btfsc STATUS,Z    ; Es un número el que voy a pronunciar?
    goto BuscaMensaISD_Num    ; Sí

    ; No
; Entonces, eso quiere decir que es una sílaba lo que voy a pronunciar
BuscaMensaISD
    nop
;    clrf    w_temp

; La primera búsqueda la hago por orden alfabético, para eso uso la primera letra de la
sílaba
    pagesel    BuscaAlfabetica
    call    BuscaAlfabetica    ; Regreso con la dirección alta y baja de la EEPROM
sílaba en sus respectivos registros

; Se lee una sílaba de la EEPROM_Sílaba... Se leen los primeros 10 bytes de la
EEPROM_Sílaba y se unen para formar las cinco letras de una sílaba escrita en la
EEPROM_Sílaba...
SteLeaCargueEEPROMSilaba
    pagesel    LeaCargueEEPROMSilaba
    call    LeaCargueEEPROMSilaba

; Ahora se compara si "Letra1 = Letra1_Sílaba", si no lo es, debe volver a leer una nueva
sílaba
    movfw Letra1_EEPROM_Silaba
    xorwf Letra1,W
    pagesel    NoEncuentraSilaba

```

```

btfss STATUS,Z ; ¿Iguales?
goto NoEncuentraSilaba ; No

```

```

; Sí
movfw Letra2_EEPROM_Silaba
xorwf Letra2,W
pagesel AjusteSteSilaba
btfss STATUS,Z ; ¿Iguales?
goto AjusteSteSilaba ; No

```

```

; Sí
movfw Letra3_EEPROM_Silaba
xorwf Letra3,W
pagesel AjusteSteSilaba
btfss STATUS,Z ; ¿Iguales?
goto AjusteSteSilaba ; No

```

```

; Sí
movfw Letra4_EEPROM_Silaba
xorwf Letra4,W
pagesel AjusteSteSilaba
btfss STATUS,Z ; ¿Iguales?
goto AjusteSteSilaba ; No

```

```

; Sí
movfw Letra5_EEPROM_Silaba
xorwf Letra5,W
pagesel AjusteSteSilaba
btfss STATUS,Z ; ¿Iguales?
goto AjusteSteSilaba ; No

```

; Sí

; Entonces se leerá (de la EEPROM_Sílaba) la dirección en la que se encuentra el mensaje en el ISD...

```
pagesel      LeaEeprom_Silaba
```

```
call      LeaEeprom_Silaba
```

```
pagesel      DeHexaADecEeprom_Silaba
```

```
call      DeHexaADecEeprom_Silaba; Pasa de Hexa a Decimal el dato leído en la
EEPROM_Sílaba
```

```
movwf ContaAltoMensaISD ; Letra1_EEPROM_Silaba ; Se guarda el dato
```

```
pagesel      Actua_Dir_EEPROM_Silaba
```

```
call      Actua_Dir_EEPROM_Silaba; Apunta al segundo byte de la
EEPROM_Sílaba (el segundo byte que contiene parte de la DirAltaISD
```

```
pagesel      LeaEeprom_Silaba
```

```
call      LeaEeprom_Silaba
```

```
pagesel      DeHexaADecEeprom_Silaba
```

```
call      DeHexaADecEeprom_Silaba
```

```
addwf ContaAltoMensaISD,1 ; Letra1_EEPROM_Silaba,1 ; Se guarda el
dato... Ya quedó guardada la "Letra1_EEPROM_Sílaba"
```

```
pagesel      Actua_Dir_EEPROM_Silaba
```

```
call      Actua_Dir_EEPROM_Silaba; ; Apunta al primer byte de la
EEPROM_Sílaba (el primer byte que contiene parte de la DirBajaISD
```

```
pagesel      LeaEeprom_Silaba
```

```
call      LeaEeprom_Silaba
```

```
pagesel      DeHexaADecEeprom_Silaba
```

```
call      DeHexaADecEeprom_Silaba
```

; Hago un enmascaramiento porque voy a convertir este número en BCD

```
movwf ContaBajoMensaISD
nop
nop
nop
nop
swapf ContaBajoMensaISD,1
nop
nop
nop
nop
```

; movwf ContaBajoMensaISD ; Letra2_EEPROM_Silaba ; Se guarda el dato

```
pagesel Actua_Dir_EEPROM_Silaba
call Actua_Dir_EEPROM_Silaba; Apunta al segundo byte de la
EEPROM_Sílaba (el segundo byte que contiene parte de la DirBajaISD
pagesel LeaEeprom_Silaba
call LeaEeprom_Silaba
pagesel DeHexaADecEeprom_Silaba
call DeHexaADecEeprom_Silaba
addwf ContaBajoMensaISD,1 ; Letra2_EEPROM_Silaba,1 ; Se guarda el
dato... Ya quedó guardada la "Letra2_EEPROM_Sílaba"
```

; Como ya tengo las direcciones (alta y baja) en la que está el mensaje, procedo a saltar entre mensajes para llegar hasta la sílaba y pronunciarla...

```
pagesel SaltaLargo1_ReproMensa_ISD ;
SaltaLargo_ReproMensa_ISD
call SaltaLargo1_ReproMensa_ISD ; SaltaLargo_ReproMensa_ISD ;
Va a buscar el mensaje entre todos los que hay en el ISD (Pronuncia la sílaba)
```

; Se actualizan direcciones para quedar apuntando a la letra1 de la siguiente sílaba en la
EEPROM_Sílaba

nop

pagesel Actua_Dir_EEPROM_Silaba

call Actua_Dir_EEPROM_Silaba; Apunta a la siguiente dirección

EEPROM_Sílaba... La primera de reserva de la sílaba

pagesel Actua_Dir_EEPROM_Silaba

call Actua_Dir_EEPROM_Silaba; Apunta a la siguiente dirección

EEPROM_Sílaba... La segunda de reserva de la sílaba

pagesel Actua_Dir_EEPROM_Silaba

call Actua_Dir_EEPROM_Silaba; Apunta a la siguiente dirección

EEPROM_Sílaba... La primera letra de la ste sílaba...

; ----- Inicio del Intento saltar entre Notas: Habilito la lectura
del RB0 como botón FFW

Anex0_11082014

; Leo el Sw_FFW

nop

pagesel Salir_Anex0_11082014

btfsc Sw_FFW ; ¿Pulsado?

goto Salir_Anex0_11082014 ; No

; Sí

movlw .200

pagesel RET_UDCMILI_20MHz

call RET_UDCMILI_20MHz

nop

pagesel Salir_Anex0_11082014

btfsc Sw_FFW ; ¿Pulsado?

goto Salir_Anex0_11082014 ; No


```

; Sí
Ste_Anex0_11082014
    nop
    pagesel    LeaEeprom
    call    LeaEeprom
    movlw .0
    xorwf    DatoRxDeEeprom,W
    pagesel    Salir_Anex0_11082014
    btfsc    STATUS,Z    ; ¿Iguales?
    goto     Salir_Anex0_11082014    ; Sí
; No

```

```

    pagesel    Actua_Dir_EEPROM
    call    Actua_Dir_EEPROM
    nop
    pagesel    Ste_Anex0_11082014
    goto     Ste_Anex0_11082014
Salir_Anex0_11082014

```

; ----- Final del Intento saltar entre Notas: Habilito la lectura
del RB0 como botón FFW

; Y ahora se regresa a continuar con la siguiente sílaba

```

    nop
    pagesel    Ste_ReproMensajesISD_xSw
    goto     Ste_ReproMensajesISD_xSw

```

AjusteSteSilaba

```

    pagesel    Actua_Dir_EEPROM_Silaba
    call    Actua_Dir_EEPROM_Silaba; Apunta al segundo byte de la Dirección Alta

```

pagesel Actua_Dir_EEPROM_Silaba
 call Actua_Dir_EEPROM_Silaba; Apunta al primer byte de la Dirección Baja
 pagesel Actua_Dir_EEPROM_Silaba
 call Actua_Dir_EEPROM_Silaba; Apunta al segundo byte de la Dirección

Baja

pagesel Actua_Dir_EEPROM_Silaba
 call Actua_Dir_EEPROM_Silaba; Apunta al primer byte de reserva de la

sílaba

pagesel Actua_Dir_EEPROM_Silaba
 call Actua_Dir_EEPROM_Silaba; Apunta al segundo byte de reserva de la

sílaba

pagesel Actua_Dir_EEPROM_Silaba
 call Actua_Dir_EEPROM_Silaba; Apunta al primer byte de la ste sílaba

pagesel SteLeaCargueEEPROMSilaba
 goto SteLeaCargueEEPROMSilaba ; Y va a cargar la ste sílaba para

volverla a comparar...

NoEncuentraSilaba

; Este es un error que se presenta en el caso de que la sílaba que viene de la
 descomposición silábica no se encuentre en la EEPROM_Silaba

; La causa de este error puede ser:

- ; 1. Error del usuario cuando digitó
- ; 2. Error del apuntador que interpretó mal la escritura del invidente/del PC
- ; 3. Error del apuntador que hizo mal la descomposición silábica
- ; 4. Error del apuntador que no tiene grabada la sílaba en cuestión (ni en la parte gramatical ni en la parte sonora)

```

pagesel    BuzzerOn
call    BuzzerOn    ; Bip en lugar de pronunciar sílaba...

```

```

;    pagesel    LedOnOff
;    call    LedOnOff

```

```

pagesel    Ste_ReproMensajesISD_xSw
goto    Ste_ReproMensajesISD_xSw    ; Va a descomponer la ste sílaba

```

```

;    pagesel    DetermNumMensaISD
;    call    DetermNumMensaISD    ; Regresa con las direcciones alta y baja para
encontrar el mensaje de la sílaba...

```

```

;    movfw DirBajaSilabaISD
nop
nop
nop
nop

```

```

pagesel    ReproMensajeISD
goto    ReproMensajeISD

```

BuscaMensaISD_Num

; Primero se direcciona hacia el número en cuestión

```

;    movfw DirAltaEeprom_L1
;    movwf DirAltaEeprom
;    movfw DirBajaEeprom_L1
;    movwf DirBajaEeprom

```

```

; Leo la Eeprom
;     pagesel     LeaEeprom
;     call      LeaEeprom

```

; Me aseguro de que los registros de trabajo de la subrutina
 "LeaActSumePRGuardeLetra" antes de invocarla por primera vez

```

;     pagesel     BorRegTrabLASPRGL
;     call      BorRegTrabLASPRGL

```

```

;Ste_HallaPRSilaba_Caso0

```

```

;     pagesel     LeaActSumePRGuardeLetra
;     call      LeaActSumePRGuardeLetra

```

```

;     movlw 0x2f
;     subwf Letra1,W
;     nop
;     nop
;     nop
;     nop
;     nop
;     movwf ContaBajoMensaISD

```

```

;     movlw .1
;     movwf ContaBajoRutSaltoLargo

```

; El sonido "CERO" está en la dirección 1, así que resto 0x29 y el resultado lo pongo en
 el "ContaBajoRutSaltoLargo"...

```

    nop
    clrf  ContaAltoMensaISD

```

```

;    movlw 0x2F
;    subwf DatoRxDeEeprom,W
;    nop
;    nop
;    nop
;    nop
;    nop
    movfw DatoRxDeEeprom
    movwf ContaBajoMensaISD

```

; Como ya tengo las direcciones (alta y baja) en la que está el mensaje, procedo a saltar entre mensajes para llegar hasta la sílaba y pronunciarla...

```

    pagesel    SaltaLargo1_ReproMensa_ISD    ;
SaltaLargo_ReproMensa_ISD
    call    SaltaLargo1_ReproMensa_ISD    ; SaltaLargo_ReproMensa_ISD    ;

```

Va a buscar el mensaje entre todos los que hay en el ISD (Pronuncia la sílaba)

```

;    movfw DatoRxDeEeprom
;    movwf PesoRelativoSílaba
;    pagesel    ReproducirISD_xSw
;    call    ReproducirISD_xSw ; Reproduce el número directamente

    pagesel    Ste_ReproMensajesISD_xSw
    goto    Ste_ReproMensajesISD_xSw

```

ReproMensajeISD

```

    nop
    nop
    nop
    nop

```

```

nop
movfw PesoRelativoSílaba ; DatoRxDeEeprom

```

```

nop

```

```

pagesel      SaltaReproMensajeISD

```

```

call  SaltaReproMensajeISD ; Aquí se va a buscar el mensaje seleccionado

```

y se va a reproducir...

```

nop

```

```

pagesel      Ste_ReproMensajesISD_xSw

```

```

goto  Ste_ReproMensajesISD_xSw

```

```

;=====

```

```

=====

```

FinApunta

```

nop

```

```

pagesel      FinApunta

```

```

goto  FinApunta ;      $-1

```

```

nop

```

```

end

```

```

;=====

```

```

=====

```